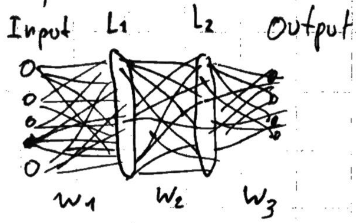
Deep learning exam

1. Basic questions

1. 

For each sample $x \in \mathbb{R}^d$ the MLP is a function

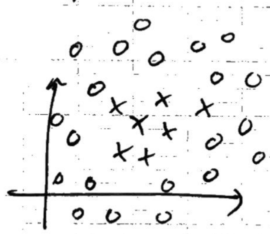
$$f: \mathbb{R}^d \longrightarrow \mathbb{R}^k \quad (k = \text{number of classes})$$

$$x \longmapsto \hat{y}$$

$$\hat{y} = W_3 z_2 + b_3$$

$$z_2 = \sigma_2(W_2 z_1 + b_2)$$

$$z_1 = \sigma_1(W_1 x + b_1)$$

2. 

The neural network is able to "bend" the space so that a linear separation correctly splits the two classes. This linear separation being $\sigma(w_{output}) > 0$

3. The derivative of an activation being a number smaller than one. Since consecutive applications of the chain rule will shrink the gradient exponentially. An activation function that solves this problem is ReLU, since its gradient is 0 or 1.

4. Dropout consists of randomly turning off some neurons during training time. More formally, if $x \in \mathbb{R}^n$ is the input of the Dropout layer, it's output will be a random variable given by ($\theta \in [0,1)$ parameter of dropout)

$$y_i = \begin{cases} 0 & \text{if } p_i < \theta \\ \frac{1}{1-\theta} x_i & \text{otherwise} \end{cases} \qquad p_i \sim U(0,1)$$

$$\frac{1}{1-\theta}$$

5.

$$\mathcal{L}_{new} = \mathcal{L} + \frac{1}{2}\lambda \|a\|^2$$

$$\nabla \mathcal{L}_{new} = \nabla \mathcal{L} + \lambda a$$

$$a^{(t+1)} = a^{(t)} - \epsilon \nabla \mathcal{L}_{new} = a^{(t)} - \epsilon\left(\nabla \mathcal{L}(a^{(t)}) + \lambda a^{(t)}\right)$$

$$= a^{(t)} - \epsilon \nabla \mathcal{L}(a^{(t)}) - \epsilon \lambda a^{(t)}$$

$$= (1 - \epsilon\lambda)\, a^{(t)} - \epsilon \nabla \mathcal{L}(a^{(t)})$$

So if λ' is the weight decay, it is equivalent to using L2 regularization with

$$\lambda = \frac{\lambda'}{\epsilon}$$

An advantage of weight decay over L2 regularization is that you control directly the strength of the decay. In normal L2, the λ factor is coupled with the learning rate.

2. Backpropagation

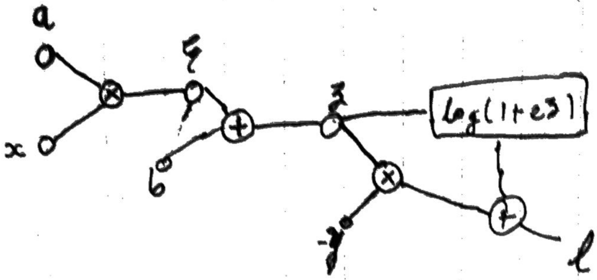
1. A computational graph is a way to express a series of computation such that it is easier to handle and manipulate. It allows us to compute derivatives effectively by means of the chain rules of the basic operations inside.

2. $$\ell(a, x, b) = y(ax+b) + \log(1 + \exp(ax+b))$$

$$\frac{\partial \ell}{\partial a} = -yx + x\,\sigma(ax+b) = x(\sigma(ax+b) + y)$$

$$\frac{\partial \ell}{\partial b} = -y + \sigma(ax+b)$$ (Using explicit derivation)

However, using back propagation we introduce the following variables:



$$l(z, y) = -yz + \log(1 + e^{z})$$

$$\frac{\partial l}{\partial z} = -y + \sigma(z) \qquad z = b + \xi \qquad \xi = ax$$

$$\frac{\partial z}{\partial b} = \frac{\partial z}{\partial \xi} = 1$$

$$\frac{\partial \xi}{\partial a} = x$$

$$\frac{\partial l}{\partial a} = \frac{\partial l}{\partial z}\frac{\partial z}{\partial \xi}\frac{\partial \xi}{\partial a} = x(\sigma(z) - y)$$

$$\frac{\partial l}{\partial b} = \frac{\partial l}{\partial z}\frac{\partial z}{\partial b} = \sigma(z) - y.$$

1. Start from the last layer, which is the output z. Compute the derivative of the loss with respect to it (we assume it's given).

2. Use this derivative to compute the derivatives of the variables used to compute the output (z)

$$\frac{\partial l}{\partial z} \longrightarrow \frac{\partial l}{\partial b}, \quad \frac{\partial l}{\partial z} \longrightarrow \frac{\partial l}{\partial \xi}$$

3. Use these new derivatives in the same fashion, until you get to the input or you get all the required derivatives (some layers may be frozen)

$$\frac{\partial l}{\partial \xi} \longrightarrow \frac{\partial l}{\partial a}$$

4. Enjoy :)

3. Because you may lose the ability to backpropagate because you need the original value. For instance, given $y = \frac{1}{2}x^2$, suppose we are provided with $\frac{\partial l}{\partial y}$ and wish to compute $\frac{\partial l}{\partial x}$. Normally, we would use $\frac{\partial l}{\partial x} = \frac{\partial l}{\partial y} \frac{\partial y}{\partial x}$ with $\frac{\partial y}{\partial x} = x$. The problem is that we don't know the value of x because it was overwritten, and if we want to get it back from y we have an ambiguity wrt the sign ($x = \pm\sqrt{2y}$). This would totally break backpropagation and gradient descent in particular (no idea where to go!). Oddly enough, the example that Caio suggested doesn't break backpropagation $y = \sin x$. Since

$$\frac{\partial y}{\partial x} = \cos x = \sqrt{1-\sin^2 x}$$
$$= \sqrt{1-y^2}.$$

So the derivative <u>can</u> be retrieved from the output.

4. The reparametrization trick is a way to perform backpropagation when a sampling operation has been done. Instead of seeing the sampling as happening inside, we consider it as a deterministic function of another random variable. This trick is essential when working with VAE.

3. Advanced Questions

1. If there is an equivariant or invariant behavior of the label with respect to translation. In the case of invariance, the convolutions should be followed by a pooling operation. An alternative approach is to augment the data keeping the same labels (invariant assumption) and train a more expressive model that will eventually learn this feature of the data.

2. The output size is $n_{out} = \left\lfloor \frac{n - n' + 1}{s} \right\rfloor$ $p_{out} = \left\lfloor \frac{p - p' + 1}{s} \right\rfloor$. We learn $K \times (n' \times p' + 1)$ weights.

We would need $(1 + n \times p) \times n_{out} \times p_{out} \times K$ (many more)

3. Images: - class of the object with respect to rotation
   - mean brightness with respect to translation/rotation

   Sound: - word in the sound with respect to intensity
   - language with respect to mean frequency.

4. An auto encoder is a neural network architecture consisting of two parts. An encoder that embeds the input in a latent space and a decoder that reconstructs the input from this latent representation. The size of the latent representation can be chosen, depending on the use case or domain. For instance, if compression is important we may want to use a small size, whereas if we care about reconstruction quality a bigger space may be necessary.

5. By using the reconstruction error. If the input is very different from the reconstruction, it may be an outlier.