Exam : Deep-Learning

Exercise 1: Multi-Layer Perceptron - 8pts

1. **(2 pts)** What is the Multi-Layer Perceptron? Provide the different components of a Multi-Layer Perceptron model (explicit the different blocks).

2. **(2 pts)** Considering a 3-class problem and $f_\theta : \mathbb{R}^n \to \mathbb{R}^3$ a neural network with $f(x^{(i)}) = \hat{y}^{(i)}$ and $y^{(i)} \in \{0, 1, 2\}$ being the index of the class associated to x_i. Provide a loss function that maximizes the log-likelihood (complete formula).

3. **(1 pt)** Write the formula of the generic gradient descent according to weights at time t (θ^t) and the loss function $\mathcal{L}(x^{(i)}, y^{(i)}, \theta^t)$.

4. **(2 pts)** What is the principle of gradient descent with momentum? Provide the formula.

5. **(1 pts)** On what conditions using l2 loss on weights have the same effect of using weight decay? Show the equivalence.

Exercise 2: The backpropagation algorithm - 9 pts

1. **(2 pts)** What structure do we use to backpropagate the gradient? On what rule/property does it rely upon? Explain briefly the mechanism?

2. **(3 pts)** Compute the gradient of the following function with respect to weights $a^{(i)}$ and $b^{(i)}$:

$$\begin{aligned} f_\theta :& \mathbb{R}^2 \to \mathbb{R} \\ & x \mapsto a^{(2)}\sigma\left(a^{(1)}.x + b^{(1)}\right) + b^{(2)} \end{aligned}$$

   With σ being the sigmoid function and $x \in \mathbb{R}^2, a^{(1)} \in \mathbb{R}^{1\times 2}$ and $a^{(2)}, b^{(1)}, b^{(2)}$ being scalars. Provide the gradient for each component.

3. **(2 pts)** Illustrate the backpropagation algorithm using the previous network and the loss $\mathcal{L}(f_\theta(x), y)$ (show what is computed in each step in a sequential order but not necessarily give the explicit gradient).

4. **(2 pts)** Provide in pseudo-code the backward function for nodes, provide the input of the function and the output, and what will be updated in the node (consequently, you need to define nodes' attributes).

Exercise 3: Deep-Learning Models - 12 pts

1. **(2 pts)** What are the benefits of using CNN architectures ? Describe briefly the different blocks of the architecture.

2. **(1 pt)** Let consider a linear/affine transformation, this transformation is applied on an image x of shape 100×100 and transforms it to a representation of shape 50×50 (reshape). How many weights are used in the transformation (order of magnitude)?

3. **(1 pt)** Let consider a convolution layer with a kernel of shape $3 \times 3 \times 1 \times 1$ (one channel in, one channel out); what is the shape of the output if we apply the convolution with a stride value of 2 and a padding of 1 on x (*error of* ± 1 *would be considered correct*)? How many weights are involved in the computation?

4. **(1 pt)** Describe the main principle of RNN architecture; for what applications such a model could be used?

5. **(2 pts)** Explain what the auto-encoder is; what are the main differences with the Variational Auto-Encoder Framework?

6. **(3 pts)** Let consider a Variational Auto-Encoder with encoder function $f_\phi : \mathbb{R}^n \to \mathbb{R} \times \mathbb{R}^+$ with $f_\phi(x^{(i)}) = (\mu^{(i)}, \sigma^{(i)})$ and g the decoder function.

   Find an expression of $-D_{KL}\left(q_\phi(z|x^{(i)})||p_\theta(z)\right)$ in term of mean $\mu_{x^{(i)}}$ and $\sigma^2_{x^{(i)}}$ the variance.

   The prior $p_\theta(z)$ is the standard normal distribution i.e. $\mathcal{N}(0, 1)$, and $q_\phi(z|x^{(i)})$ is a normal distribution. Provide a simplified formulation getting rid of the sum (with no expected value symbol).

7. **(2 pts)** What is the reparameterization trick? Explain how z is sampled ($z \sim q_\phi(z|x^{(i)})$)? What would be the gradient $\nabla_{\sigma,\mu}(\mathcal{L}(\phi, \theta, x))$? Write it according to $\nabla_z \mathcal{L}(\phi, \theta, x)$.

Cheat-Sheet

The Kullback Liebler divergence. Let consider two distribution p and q the Kullback Liebler divergence of p with q is given by:

$$D_{KL}(p||q) = \mathop{\mathbb{E}}_{x\sim p(x)} \log\left(\frac{p(x)}{q(x)}\right) = \int p(x)\log\left(\frac{p(x)}{q(x)}\right)dx$$

Gaussian PDF. The probability p having x following a gaussian distribution (1-dimensional) $\mathcal{N}(\mu, \sigma^2)$ is given by:

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\frac{(\mu-x)^2}{\sigma^2}}$$

Softmax.

Let be $x \in \mathbb{R}^n$:

$$Softmax(x_i) = \left(\frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}}\right)$$

Variance.

$$\mathop{\mathbb{E}}_{x\sim p(x)}[x^2] = \sigma_x^2 + \mu_x^2$$
$$\mathop{\mathbb{E}}_{x\sim p(x)}[(x-\mu_x)^2] = \sigma_x^2$$

Sigmoid activation.

./sigmoid_with_gradient.pdf

The sigmoid formula and its derivative:

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{1}$$
$$\sigma(x)' = \sigma(x)(1-\sigma(x)) \tag{2}$$