Exercises
Deep Learning
Fall 2018

Machine Learning Institute
Dept. of Computer Science, ETH Zürich
Fernando Pérez-Cruz
Web http://www.da.inf.ethz.ch/teaching/2018/DeepLearning/

Series Monday, Oct 15, 2018
(Deep Learning, Exercise series 3)

Problem 1 (Activation Functions):

1. Consider the activation function
$$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0. \end{cases} \tag{1}$$
Why is (1) generally not used for neural network training?

2. Now, consider a two-layer feedforward network in which the non-linear activations are given by the sigmoid function $\sigma(z) = \frac{1}{1+\exp(-z)}$. Show that there exists an equivalent network, which computes exactly the same function, but with activations given by $\tanh(z) = \frac{\exp(z)-\exp(-z)}{\exp(z)+\exp(-z)}$. First derive the relation between $\sigma(z)$ and $\tanh(z)$ and then show that the parameters of the two networks differ by linear transformations.

Problem 2 (Elementary Logic Functions):

Feedforward neural networks with linear activations (i.e. identity activation functions) have many limitations. Most famously, they cannot learn the XOR($[x_1, x_2]$) function: XOR($[0,1]$) $= $ XOR($[1,0]$) $= 1$ and XOR($[0,0]$) $=$ XOR($[1,1]$) $= 0$.

1. Give a short proof or illustrate in the x_1, x_2-plane why this is not possible.

2. Show that a non-linear two-layer neural network can in fact solve the XOR problem. Consider the network given by $f(\mathbf{x}; \mathbf{W_2}, b_2, \mathbf{W_1}, \mathbf{b}_1) = \mathbf{W}_2\, h(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1) + b_2$ with $h(z) = \max(0, z)$ and
$$\mathbf{W}_1 = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \quad \mathbf{b}_1 = \begin{bmatrix} 0 \\ -1 \end{bmatrix}, \quad \mathbf{W}_2 = ?, \quad b_2 = ?$$
Find the corresponding weights $\mathbf{W}_2$ and bias b_2 and show that this network gives the correct output on all inputs $\mathbf{x} \in \{0,1\}^2$.

Problem 3 (Gradients of the Common Neural Network Layers):

The generalization of derivative to high order functions $f : \mathbb{R}^n \to \mathbb{R}^m$ is the Jacobian[1] which is a $m \times n$ matrix of partial derivatives. This is a central notion used in deep learning and it will be extensively used when deriving the back-propagation algorithm.

In this exercise we will compute Jacobians of some common neural network layers.

1. $x \in \mathbb{R}^n, f(x) = ReLU(x) = \max(x, 0)$. Compute $\frac{\partial f}{\partial x}$.
2. $x \in \mathbb{R}^n, f(x) = HardTanh(x)$. Compute $\frac{\partial f}{\partial x}$.
3. Max layer: $x \in \mathbb{R}^n, f : \mathbb{R}^n \to \mathbb{R}, f(x) = \max_i(x_i)$. Compute $\frac{\partial f}{\partial x}$.
4. Element-wise multiplication layer: $x, \theta \in \mathbb{R}^n, f : \mathbb{R}^n \to \mathbb{R}^n, f(x) = x \odot \theta$. Compute $\frac{\partial f}{\partial x}$.

[1] `https://en.wikipedia.org/wiki/Jacobian_matrix_and_determinant`

5. Linear layer: $x \in \mathbb{R}^n, W \in \mathbb{R}^{m\times n}, b \in \mathbb{R}^m, f : \mathbb{R}^n \to \mathbb{R}^m, f(x) = Wx + b$. Compute $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial b}$. If $m = 1$ compute also $\frac{\partial f}{\partial W}$, which for higher m would have been a 3-rd order tensor, but about this we'll talk another time:).

6. Softmax layer: $x \in \mathbb{R}^n, f : \mathbb{R}^n \to \mathbb{R}^n, f(x) = \text{log-softmax}(x) = \log(\text{softmax}(x))$, where $\text{softmax(x)}_i = \frac{\exp(x_i)}{\sum_j \exp(x_j)}$. Compute $\frac{\partial f}{\partial x}$. This is the typical final neural network layer used when doing classification.

Problem 4 (Derivation Gradient Descent):

Let $n \geqslant 2$. Let $f \in \mathcal{C}_b^2(\mathbb{R}^n, \mathbb{R})$ (twice differentiable, with continuous and bounded second order partial derivatives), $x_0 \in \mathbb{R}^n$ and $\varepsilon > 0$ small enough (you will have to justify how small). We set $x_0^\varepsilon = x_0$, and for all $k \in \mathbb{N}$,

$$x_{k+1}^\varepsilon = \arg\min_{x\in\mathbb{R}^n} [f(x) + \frac{1}{2\varepsilon}\|x - x_k^\varepsilon\|^2].$$

1. Justify that if ε is small enough, then the sequence $(x_k^\varepsilon)_{k\in\mathbb{N}}$ is well defined, *i.e.* show the existence and uniqueness of the minimum of the function $f + \frac{1}{2\varepsilon}\| \cdot - x_k^\varepsilon\|^2$ for some $\varepsilon > 0$ small enough, in a way that is independent of k. *Hint:* for the uniqueness, you can use that the minimum of a strictly convex function, when it exists, is unique.

2. Prove that for all $k \in \mathbb{N}$,

$$\frac{1}{\varepsilon}(x_{k+1}^\varepsilon - x_k^\varepsilon) = -\nabla f(x_{k+1}^\varepsilon).$$

   Remark. We can show a continuous version of this result. Let x^ε be the unique continuous function from $[0, 1]$ to $\mathbb{R}$ such that $x^\varepsilon(k\varepsilon) = x_k^\varepsilon$ and x^ε is linear on $[k\varepsilon, (k + 1)\varepsilon]$ (we can suppose that ε is of the form $\frac{1}{N}$ for some $N \in \mathbb{N}^*$, so that x^ε is well defined on $[1 - \varepsilon, 1]$). Then, as $\varepsilon \to 0$, the sequence of functions $(x^\varepsilon)_{\varepsilon>0}$ converges uniformly to the unique solution of the differential equation

$$x' = -\nabla(f)(x),$$

   which is the continuous path obtained by *gradient descent from initialization* x_0, f being the function we try to minimize, ε playing a role that is similar to the one played by the learning rate.

Problem 5 (Local quadratic approximation):

Consider a neural network with parameters $\boldsymbol{w} \in \mathbb{R}^d$ and a loss function $f(\boldsymbol{w})$ (such as the cross-entropy loss commonly used to train neural networks).

We can approximate $f(\boldsymbol{w})$ using a Taylor expansion around $\bar{\boldsymbol{w}}$:

$$f(\boldsymbol{w}) \approx f(\bar{\boldsymbol{w}}) + (\boldsymbol{w} - \bar{\boldsymbol{w}})^\top \nabla f(\bar{\boldsymbol{w}}) + \frac{1}{2}(\boldsymbol{w} - \bar{\boldsymbol{w}})^T H(\boldsymbol{w} - \bar{\boldsymbol{w}}), \tag{2}$$

where $H \in \mathbb{R}^{d\times d}$ is the Hessian matrix evaluated at $\bar{\boldsymbol{w}}$. Note that we ignore the cubic and higher terms in the expansion.

We write λ_i and $\boldsymbol{u}_i$ the eigenvalues and eigenvectors of H, i.e.

$$H\boldsymbol{u}_i = \lambda_i \boldsymbol{u}_i, \tag{3}$$

where the eigenvectors $\boldsymbol{u}_i$ are orthonormal.

Let $\boldsymbol{w}^*$ be a minimum of the error function (i.e. $\boldsymbol{w}^* \in \operatorname{argmin}_{\boldsymbol{w}} f(\boldsymbol{w})$). We now expand $\boldsymbol{w} - \boldsymbol{w}^*$ as a linear combination of the eigenvectors, i.e.

$$\boldsymbol{w} - \boldsymbol{w}^* = \sum_i \alpha_i \boldsymbol{u}_i \tag{4}$$

This can be regarded as a transformation of the coordinate system in which the origin is translated to the point $\boldsymbol{w}^*$, and the axes are rotated to align with the eigenvectors.

1. Write down the Taylor expansion of $f(\boldsymbol{w})$ at $\bar{\boldsymbol{w}} = \boldsymbol{w}^*$.
2. Show that
$$f(\boldsymbol{w}) \approx f(\boldsymbol{w}^*) + \frac{1}{2}\sum_i \lambda_i \alpha_i^2. \tag{5}$$

Problem 6 (Weierstrass theorem):

In this exercise, we seek to derive a formal proof of the Weierstrass theorem discussed in the lecture. Recall that the theorem can be stated as follows:

Theorem 1. *If $f(x)$ is a given continuous function for $a \leq x \leq b$ and if ϵ is an arbitrary positive quantity, it is possible to construct an approximating polynomial $P(x)$ such that*

$$|f(x) - P(x)| \leq \epsilon, \quad a \leq x \leq b \tag{6}$$

Without loss of generality we assume $0 < a < b < 1$ and $f(x) = 0$ outside the interval (a, b).

Let $P_n(x)$ be a polynomial of degree $2n$ such that

$$P_n(x) = \frac{1}{J_n}\int_0^1 f(t)[1-(t-x)^2]^n \, dt, \tag{7}$$

where J_n is the constant $J_n = \int_{-1}^{1}(1-u^2)^n \, du$.

1. Show that
$$f(x) = \frac{1}{J_n}\int_{-1}^{1} f(x)(1-u^2)^n \, du \tag{8}$$
2. Show that
$$P_n(x) - f(x) = \frac{1}{J_n}\int_{-1}^{1}[f(x+u) - f(x)](1-u^2)^n \, du \tag{9}$$
The problem is now to show that this expression approaches zero as $n \to \infty$.
3. Let $\epsilon > 0$. Since $f(x)$ is continuous there exists a $\delta > 0$ such that $|f(x+u) - f(x)| \leq \frac{\epsilon}{2}$ for each u small enough, $|u| < \delta$. Show that
$$|f(x+u) - f(x)| \leq \frac{\epsilon}{2} + 2M\frac{u^2}{\delta^2}, \tag{10}$$
where $|f(x)| \leq M \;\; \forall x \in [a-1, b+1]$.
Hint: Think of the case $|u| \geq \delta$, i.e. $1 \leq \frac{u^2}{\delta^2}$.
4. Using integration by part, show that $J_n' := \int_{-1}^{1} u^2(1-u^2)^n \, du = \frac{J_{n+1}}{2(n+1)}$
5. Finally, re-using the answers to the previous questions, prove that
$$|f(x) - P_n(x)| \leq \epsilon \tag{11}$$
for sufficiently large n.

Problem 7 (Weierstrass theorem, simplification in the $\mathcal{C}^\infty$ case):

Let $a, b \in \mathbb{R}$ with $a < b$. Let $f \in \mathcal{C}^\infty([a,b], \mathbb{R})$ such that

$$\exists q \in \mathbb{N} \mid \forall n \in \mathbb{N}, \; \|f^{(n)}\|_\infty = O_{n\to\infty}(q^n), \tag{12}$$

where the sup norm $\|\cdot\|_\infty$ is taken over $[a,b]$. (Intuitively, that is to say that the successive partial derivatives of f don't grow faster than all geometric sequences.) Give a short proof of Weierstrass' approximation theorem, *i.e.* there exists $(P_n)_{n\in\mathbb{N}} \in (\mathbb{R}[X])^{\mathbb{N}}$ such that $\|f - P_n\|_\infty \to_{n\to\infty} 0$.