Deep Learning

Instructor: Dr. Davoodabadi Farahani
Head TA: Ali Momen

Semester: Fall 2025

Homework 1:

Introduction to Deep Learning

Designed By:

Amin Noormahmoodi
@a_m82i_n

Sepehr Razavi
@s_sepehr_razavi

Deadline: 23 Mehr

Solution Release & Presentation Day: 1-2 Aban

Preface

In this homework you will gain further intuition regarding the concepts presented in the first six slides from MLPs to the methods for better training the neural nets.

Notes and Honor Code

This homework is part of the Deep Learning course at IUST offered in Fall 2025 by Dr.Davoodabadi. Please read all instructions carefully before starting.

- Collaboration is encouraged, but each student must submit their own work.
- Submitting other students' work or copying solutions from another student would result in a 0 score on the assignments
- Typesetting in LaTeX is strongly recommended.
- Clearly mention collaborators, if any.

If you have any further questions regarding the submission policies,course policies e.t.c. feel free to contact the Head Teaching Assistant of the course, Ali Momen, via Telegram.

Submission

The **soft deadline** deadline for this homework is **23 Mehr**.
Please submit your work by following the instructions below:

* Place your solution alongside the Jupyter notebook(s). Your written solution must be a single PDF file named **HW1_Solution.pdf**.

* Zip all the files together with the following naming format:
  DL_HW1_[StudentNumber]_[FullName].zip.
  Replace `[FullName]` and `[StudentNumber]` with your full name and student number, respectively. Your `[FullName]` must be in CamelCase with no spaces.

* Submit the zip file through Quera in the appropriate section.

* Please note that you may use up to **7 slack days** for this assignment.

1 Theoretical Problems

Problem 1. Optimizers(15 points) You are training a deep neural network for image classification. To improve convergence, you are comparing several gradient-based optimization algorithms. Answer the following questions:

(a) Explain the main idea behind the **Momentum** optimization method. How does it improve upon standard gradient descent?

(b) Describe the difference between **Momentum** and **Nesterov Accelerated Gradient (NAG / Nesterov momentum)**.

(c) The **AdaGrad** optimizer adapts learning rates during training. Explain how it does this, and discuss one major limitation of AdaGrad.

(d) How does **RMSProp** modify the idea of AdaGrad to address its limitation?

(e) The **Adam** optimizer combines ideas from momentum and RMSProp. Describe how Adam works.

Problem 2. Kaiming He(10 points) In the lectures, you have seen that it is common to initialize the weights of a neural net that has zero-centered activation functions using the Xavier method which tries to mitigate the output of activations becoming too small. Since this method assumes the activation to be zero-centered, it does not work as expected with the ReLu activation function. In this problem, we try to derive a method for properly initializing weights for ReLu activation.
Consider an H-layer fully-connected feedforward neural network with ReLU activations:

$$x^{(0)} = x \in \mathbb{R}^{d_0}, \qquad z^{(h)} = W^{(h)} x^{(h)} \in \mathbb{R}^{d_{h+1}}, \qquad x^{(h+1)} = \sigma\big(z^{(h)}\big),$$

for $h = 0, 1, \ldots, H-1$, where $\sigma(t) = \max\{0, t\}$ is the ReLU applied elementwise. Index components by $x_j^{(h)}$ and $z_i^{(h)}$. Assume for a fixed layer h the weight entries $\{W_{ij}^{(h)}\}_{i,j}$ are i.i.d. with

$$\mathbb{E}[W_{ij}^{(h)}] = 0, \qquad \mathrm{Var}(W_{ij}^{(h)}) = \sigma_h^2,$$

and that weights are independent of activations from previous layers.

(a) **Zero mean.** Show that for every i,

$$\mathbb{E}\big[z_i^{(h)}\big] = 0.$$

(b) **Variance expansion.** Write $z_i^{(h)} = \sum_{j=1}^{d_h} W_{ij}^{(h)}\, x_j^{(h)}$. Using independence of the terms across j, derive an expression for $\mathrm{Var}\big(z_i^{(h)}\big)$ in terms of σ_h^2 and the second moment $\mathbb{E}\big[(x_j^{(h)})^2\big]$. In particular show that

$$\mathrm{Var}\big(z_i^{(h)}\big) \;=\; d_h\, \sigma_h^2\, \mathbb{E}\big[(x_j^{(h)})^2\big].$$

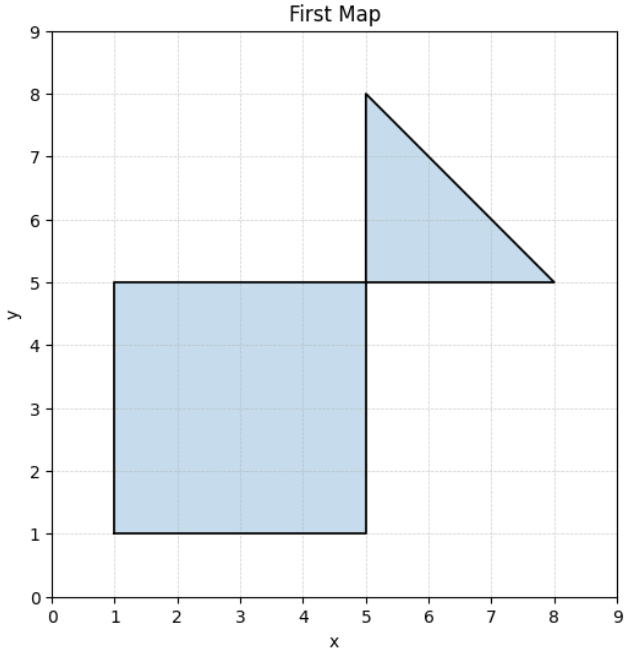
(c) **ReLU second moment.** Assume that the distribution of $z_j^{(h-1)}$ is symmetric about zero (i.e. its density satisfies $p(z) = p(-z)$) and has mean zero. Prove that

$$\mathbb{E}\big[(x_j^{(h)})^2\big] \;=\; \tfrac{1}{2}\,\mathrm{Var}\big(z_j^{(h-1)}\big).$$

(Hint: write the integral for $\mathbb{E}[(\max\{0, z\})^2]$ and use symmetry.)

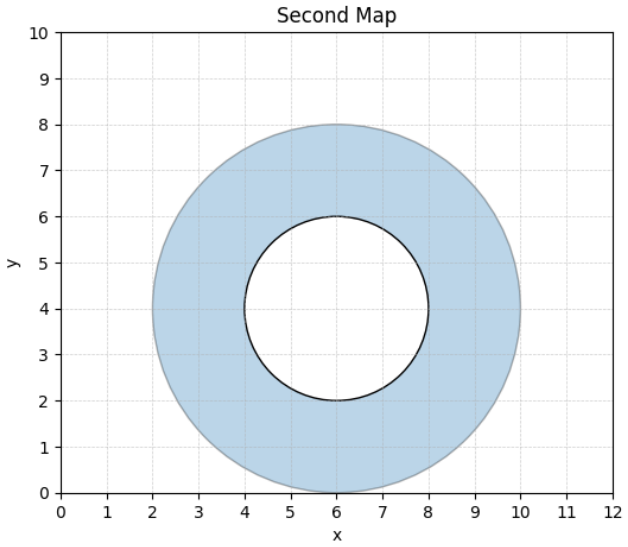
(d) **Combine results.** By using the previous results derive a relation for σ_h^2 in order to preserve the variance of the inputs across the layers.

Problem 3. War Zone(10 points) A car is moving inside a special region that once was a war zone. There are still some areas in this region that are known to be dangerous due to the probable presence of bombs hidden under the ground. The officials of this region has provided different maps, determining the areas that are known to be dangerous. For each of the following maps, design a separate system that starts off an alarm to notify the driver of the car of the imminent danger.



(a)

(b) (Hint: You can think of features non-linearly!)



Problem 4. Divergence(10 points) In this question we want to analyze the effect of learning rate in single variable quadratic function with more detail. Suppose that we have quadratic function with the following form:

$$E = \frac{1}{2}aw^2 + bw + c$$

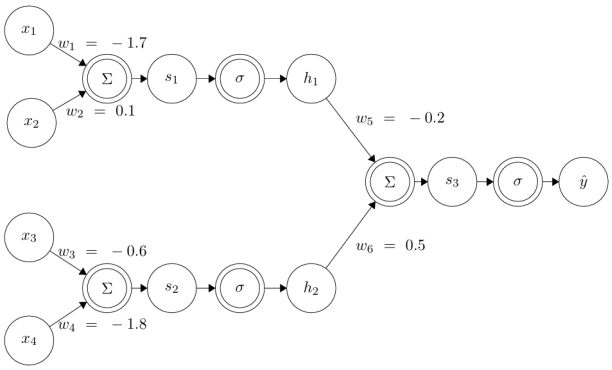
(a) By using the Taylor expansion show that the second order derivative is the optimal learning rate, η_{opt}, for the gradient decent algorithm.

(b) Propose a function that its second-order derivative is not the optimal learning rate.

(c) Now prove that in quadratic functions with a positive second-order derivative ($a > 0$), if the learning rate becomes greater than twice the optimal learning rate ($\eta > \eta_{opt}$), algorithm would diverge. (Explanation: We call an algorithm diverging, when the difference of the parameter w with its optimal value w^* increases in each step of the algorithm.)

(d) Provide an intuitive explanation for the oscillating convergence of the algorithm when $2\eta_{opt} > \eta > \eta_{opt}$?

Problem 5. Backpropagation(15 points) Consider the following neural network. Single-circled nodes denote variables (e.g. x_1 is an input variable, h_1 is an intermediate variable, $\hat{y}$ is an output variable), and double-circled nodes denote functions (e.g. Σ takes the sum of its inputs, and σ denotes the logistic function $\sigma(x) = \frac{1}{1+e^{-x}}$). In the network below, $h_1 = \frac{1}{1+e^{-x_1 w_1 - x_2 w_2}}$. Suppose we have an L2 loss $L(y, \hat{y}) = \|y - \hat{y}\|^2$. We are given a data point $(x_1, x_2, x_3, x_4) = (-0.7, 1.2, 1.1, -2)$ with true label 0.5. Use the backpropagation algorithm to compute the partial derivative $\frac{\partial L}{\partial w_1}$.



Problem 6. Dropout(10 points) Here we explore the effect of dropout regularization on a simple linear regression model trained using least squares. Consider a model of the form

$$y_k = \sum_{i=1}^{D} w_{ki} x_i$$

along with a sum-of-squares error function given by

$$E(\mathbf{W}) = \sum_{n=1}^{N} \sum_{k=1}^{K} \left\{ y_{nk} - \sum_{i=1}^{D} w_{ki} R_{ni} x_{ni} \right\}^2$$

where the elements $R_{ni} \in \{0, 1\}$ of the dropout matrix are chosen randomly from a Bernoulli distribution with parameter ρ.

(a) Show that

$$\mathbb{E}[R_{ni}] = \rho$$

$$\mathbb{E}[R_{ni} R_{nj}] = \delta_{ij}\rho + (1 - \delta_{ij})\rho^2.$$

(b) Now show that the following is the expectation of the error function defined above:

$$\mathbb{E}\left[E(\mathbf{W})\right] = \sum_{n=1}^{N}\sum_{k=1}^{K}\left\{y_{nk} - \rho\sum_{i=1}^{D} w_{ki}x_{ni}\right\}^2 + \rho(1-\rho)\sum_{n=1}^{N}\sum_{k=1}^{K}\sum_{i=1}^{D} w_{ki}^2 x_{ni}^2.$$

(c) What does the second term of the above expectation resemble?

2 Coding challenges

2.1 Practical-Pytorch Notebook (30 points)

In this notebook, you will learn the basics of PyTorch, a widely used deep learning framework for this course. Additionally, you will gain hands-on experience with key concepts such as training a model and performing backpropagation through a computation graph. Finally, you will build a model for the Continuous XOR dataset, evaluate its performance, and visualize its decision boundaries.

Complete all the **TODO** sections in the notebook to obtain the correct results and achieve a full score. The deliverable for this part is **one completed notebook with results**.

Good Luck!!!

References

[1] Christopher M. Bishop and Hugh Bishop. *Deep Learning: Foundations and Concepts*. Springer, 2024.

[2] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. Book in preparation for MIT Press.