NNDL2025: Exercises & Assignments
Session 1 (20 Mar)

1 Basic definitions

Mathematical exercises

1. (8 pts) One proposal as activation function is the ”leaky ReLU”:

$$\psi(x) = \begin{cases} x, & \text{if } x \geq 0 \\ \alpha x, & \text{if } x < 0 \end{cases} \tag{1}$$

for some constant $0 \leq \alpha \leq 1$, typically small.

(a) Show that the basic ReLU is a special case of this.

(b) Show that the linear activation function is also a special case.

(c) Show that the leaky ReLU can also be expressed as:

$$\psi(x) = \max(\alpha x, x) \tag{2}$$

2. (8 pts) Take a multi-layer neural network with linear activation function:

$$\mathbf{y}_K = \mathbf{W}_K \mathbf{W}_{k-1} \dots \mathbf{W}_2 \mathbf{W}_1 \mathbf{x} = \mathbf{M}\mathbf{x} \;\; =: \mathbf{g}(\mathbf{x}) \tag{3}$$

where the total function given by the NN is denoted by $\mathbf{g}$.

(a) Suppose we know that $\mathbf{W}_1$ is of size $m \times n$. What kind of constraint does this imply on the size of $\mathbf{W}_2$?

(b) Suppose again that we know that $\mathbf{W}_1$ is of size $m \times n$. What is a necessary condition on m, n to make it possible that the neural network is injective (also called invertible in NN literature), meaning that for any output $\mathbf{y}$ you can compute the original $\mathbf{x}$. (A non-rigorous answer is fine.)

(c) Assume that all matrices $\mathbf{W}_i$ are square (same number of rows and columns). In the case of an arbitrary number of layers, give a necessary and sufficient condition on the $\mathbf{W}_i$ such that this $\mathbf{g}$ is injective.

Computer assignments

1. (9 pts) Construct a neural network using PyTorch. As architecture, take a fully connected neural network with three layers of weights, and five neurons in each layer. (Thus, the input and the output are also 5-dimensional.) Try out the tanh, ReLU, and linear activation functions, always the same activation everywhere in the NN.

2. (9 pts)Create three sets of random weight vectors (for different plots to be made). Plot y_1, that is the first entry of the output vector, as a function of x_1 from -10.0 to 10.0, when all the other $x_i, i > 1$ are fixed to random values. **Report** three plots (corresponding to the three sets of random weights) for each of the three activation functions; in total, nine plots.

2 Optimization

Mathematical exercises

In the following exercises, when the exercise says "calculate", it means you should show the detailed derivation.

We denote by $\mathbf{w}$ an n-dimensional vector.

1. (7 pts) Calculate the gradient of the function $f_1(\mathbf{w}) = \|\mathbf{w}\|^2$. Hint: write out the function so that you express it as a function of the individual entries w_i; then take the partial derivatives as in the definition of a gradient.

2. (7 pts) Calculate the Hessian of the function $f_1(\mathbf{w}) = \|\mathbf{w}\|^2$. Hint: start from the result of the preceding.

3. (7 pts) Derive Newton's method for optimizing $f_1(\mathbf{w}) = \|\mathbf{w}\|^2$. It works incredibly well; why?

4. (7 pts) Calculate the gradient of the function $f_0(\mathbf{w}) = \mathbf{w}^T\mathbf{z}$ for some fixed vector $\mathbf{z}$.

5. (5 pts) Calculate the gradient of the function

$$f_2(\mathbf{w}) = g(\mathbf{w}^T\mathbf{z}) \tag{4}$$

for some fixed (non-stochastic) vector $\mathbf{z}$. Here, g is some differentiable function $\mathbb{R} \to \mathbb{R}$. Hint: as always, write out the dot-product so you have the function of the individual entries w_i.

6. (5 pts) Calculate the *stochastic* gradient of the function

$$f_3(\mathbf{w}) = \mathrm{E}\{g(\mathbf{w}^T\mathbf{z})\} \tag{5}$$

for some *random* vector $\mathbf{z}$.

7. (2 bonus pts) A couple of short bonus questions, possibly fun, on gradients. Here, the point is to further illustrate how the calculation of the gradients obeys formally similar rules as elementary univariate derivatives. In these bonus questions, don't make any detailed calculations, just use common sense and pay attention to the dimension of the result; no justification needed for your answer.

   (a) We know that the derivative of $\frac{1}{2}ax^2$ is ax. Which of the following is the gradient of $f(\mathbf{w}) = \frac{1}{2}\mathbf{w}^T\mathbf{M}\mathbf{w}$ where $\mathbf{M}$ is some symmetric matrix?

      i. $\mathbf{M}$

      ii. $\mathbf{w}^T\mathbf{M}$

      iii. $\mathbf{M}\mathbf{w}$

      iv. $\mathbf{M}\mathbf{w}^T$

   (b) We know that the derivative of $\frac{1}{4}x^4$ is x^3. Which of the following is the gradient of $f(\mathbf{w}) = \frac{1}{4}\|\mathbf{w}\|^4$?

      i. $\mathbf{w}\mathbf{w}\mathbf{w}$

      ii. $\mathbf{w}^T\mathbf{w}\mathbf{w}^T$

      iii. $\|\mathbf{w}\|^2\mathbf{w}$

      iv. $\|\mathbf{w}\|^3$

Computer assignments

You should do the calculations of the gradients by hand and implement them in numpy.

1. (28 pts) Consider the function $f : \mathbb{R}^2 \to \mathbb{R}$:

$$f(\mathbf{w}) = \exp(-w_1^2 - 2(w_2 - 1)^2) + \exp(-(w_1 - 1)^2 - 2w_2^2). \qquad (6)$$

   (a) To begin with, plot the isocontours of the function f (use your implementation of $f(\mathbf{w})$, which takes in an array and output a scalar)

   (b) Calculate the gradient of f

   (c) Maximize this with the gradient method. Start with three "random" initial points which we define here as $(0.2, 0.5), (0.5, 0.2), (1.0, 1.0)$ for convenience. Note: You have to try different step sizes to find a good one. (Hint: you may have to try wildly different step sizes of completely different magnitudes to find a good one). Use some stopping criterion given in the lecture material; as the threshold something like 10^{-4} might work well.

   (d) Plot the trajectories for the three runs corresponding to the three initial points above, using the best step size you found (doesn't have to be the the same for all initial points). More precisely, show the points $\mathbf{w}_i$ on the 2D plane where i is the iteration count. Joining the points for each run using

```
ax.arrow(x, y, dx, dy,
length_includes_head=True, width=0.03)
```

for better visualization. (Note: the above command draws an arrow from (x, y) to $(x + dx, y + dy)$; replace them with the correct variables.) **Report** One figure with all the three trajectory plots side by side. Each subplot corresponds to an initial point. *Overlay* the optimization trajectory on the contour plot of f from (a).

(e) Plot the objective function f as a function of iteration count. **Report** one figure in which all the three curves are shown side by side.

(f) **Report** a brief discussion on: What can you conclude from the above in terms of local vs. global maxima?