CSC413 Math Homework 2

Deadline: March 11, 2024 by 5pm

Submission: Compile and submit a PDF report containing your written solutions and your code. You may also submit an image of your legible hand-written solutions. Submissions will be done on `Markus`.

Late Submission: Please see the syllabus for the late submission criteria.

Collaboration Policy: Please see the syllabus for the collaboration policy.

Question 1. Dead Units [15 Points]

Consider the following neural network, where $\mathbf{x} \in \mathbb{R}^{\mathbf{2}}$ and $\mathbf{h} \in \mathbb{R}^{\mathbf{2}}$, and $y \in \mathbb{R}$:

$$\mathbf{h} = \text{ReLU}(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}),$$
$$y = \mathbf{w}^{(2)}\mathbf{h} + b^{(2)}.$$

Suppose also that each element of $\mathbf{x}$ is between -1 and 1.

a) [5pts] Come up with values of the parameters $\mathbf{W}^{(1)}$ and $\mathbf{b}^{(1)}$ such that both hidden units h_1 and h_2 are dead, that is, their outputs are zero.

b) [5pts] Show that the gradients of y with respect to $\mathbf{W}^{(1)}$ and $\mathbf{b}^{(1)}$ are zero.

c) [5pts] Answer one of following:

- Suppose weight decay applies to weights only; what domain of initial values of

$$\mathbf{b}^{(1)}$$

would allow for weight decay to eventually revive these dead neurons in general?
- Suppose both weights and biases are penalized; show that weight decay alone cannot revive these dead neurons.

Question 2: Optimization with Adaptive Gradient Method [10 Points]

Recall the update rule of RMSProp:

$$s(k+1) \leftarrow \beta_2 s(k) + (1-\beta_2)\left[\frac{\partial \mathcal{E}(\theta(k))}{\partial \theta}\right]^2$$
$$\theta(k+1) \leftarrow \theta(k) - \frac{\alpha}{\sqrt{s(k+1)+\epsilon}}\frac{\partial \mathcal{E}(\theta(k))}{\partial \theta},$$

where the operations are performed component-wise: $(\frac{\partial \mathcal{E}(\theta_k)}{\partial \theta})^2$ is a vector with its j-th dimension being $(\frac{\partial \mathcal{E}(\theta_k)}{\partial \theta_j})^2$, and likewise for the division.

Suppose that the magnitude of $\frac{\partial \mathcal{E}(\theta(k))}{\partial \theta_1}$ tends to be large, but the magnitude of $\frac{\partial \mathcal{E}(\theta(k))}{\partial \theta_2}$ tends to be small across a large range of iterations k.

- What is the effect of using RMSProp rather than (S)GD on the trajectory of the parameters θ_1 and θ_2? We are looking for an explanation of why, for example, one of the θs will change faster/slower due to using RMSProp, and whether that's a good thing.

Question 3: Dropout [10 Points]

In a dropout layer, instead of "zeroing out" activations at test time, we multiply the weights by $1-p$, where p is the probability that an activation is set to zero during training.

Explain why the multiplication by $1-p$ is necessary for the neural network to make meaningful predictions.

Question 4: Dynamics of Learning in a Deep Neural Network [60 Points]

We know that the choice of the learning rate can affect the performance of a neural network. We would like to study it closely in a simple regression problem with a simple deep linear neural network. This question has a combination of mathematical derivations and numerical simulations, so you need to derive some mathematical results, write some codes, which should be submitted too, and produce some visualizations.

The target function of our regression problem is the function $f^*(x) = 0$. We essentially want to see if we can learn a zero function.

As of the model, we consider a simple linear deep neural network that is defined as

$$f(x; w_1, w_2) = w_2 w_1 x,$$

with $x, w_1, w_2 \in \mathbb{R}$. That is, the input, the size of the first hidden layer, and the output are all 1D, and the activation function is linear. Obviously, this is equivalent to a linear model vx with $v = w_1 w_2$, but suppose that we treat it as a deep NN.

To fully define the problem, we need to specify the distribution of the input X. To simplify our calculations, assume that X comes from a zero mean distribution with a variance of 1, that is $\mathbb{E}[X] = 0$ and $\mathbb{E}[X^2] = 1$.

Part (a) – Basic Properties of the Loss Function [30 Points]

Recall that the squared error loss of predicting y when the target is t is $l(y, t) = \frac{1}{2}|y - t|^2$.

1. [5pts] Write down the pointwise squared error loss $l(f(x; w_1, w_2), f^*(x))$ of the model $f(x; w_1, w_2)$ for the target $f^*(x)$.
2. [5pts] Write down the population loss of the model, that is $L(w_1, w_2) = \mathbb{E}[l(f(X; w_1, w_2), f^*(X))]$. Your solution should only be a function of w_1 and w_2, and not X.
3. [5pts] Compute the gradient of the population loss with respect to parameters (w_1, w_2).
4. [5pts] Compute the Hessian of the population loss with respect to parameters (w_1, w_2).
5. [5pts] Is the population loss convex or not? Prove it. [Hint: Can you test convexity of a function based on its Hessian?]
6. [5pts] What is the minimum of $L(w_1, w_2)$? What is the set of its minimizers?

Part (b) – Gradient Update [5 Points]

We use GD to train this network. We use a per-layer learning rates α_1 and α_2. We initialize the weights as $w_1(0), w_2(0)$ and update the weights for $k \geq 0$ as follows

$$w_1(k+1) \leftarrow w_1(k) - \alpha_1 \frac{\partial L(w_1(k), w_2(k))}{\partial w_1},$$
$$w_2(k+1) \leftarrow w_2(k) - \alpha_2 \frac{\partial L(w_1(k), w_2(k))}{\partial w_2}.$$

Write down the update rule in an explicit form based on $L(w_1, w_2)$ you obtained above.

Part (c) – Effect of Learning Rate [25 Points]

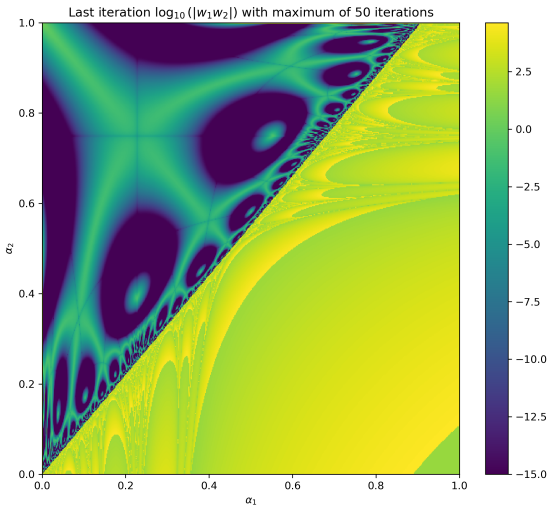
We would like to understand the dynamics of learning and the effect of learning rates on it. For example, we want to know if a certain choice of learning rates converges to the minimizer of $L(w_1, w_2)$, and if it does, how fast that happens.

If the network was only a single layer one (that is, $f(x; w) = wx$), we could use the analysis discussed in the lecture (using eigenvalues, etc.) to understand the effect of learning. But since this is a deep NN, albeit a simple one, the exact mathematical analysis is more complicated. So instead of using our math skills, we take a numerical approach to study the same questions.

You need to write some code for the rest of this question. Your code should perform GD with the update rule above starting from some initial weights and report the loss $L(w_1(k), w_2(k))$ after k iterations. Specifically, you need to generate figures where the X-axis sweeps over various values of α_1 and Y-axis sweeps over α_2 and the colour shows

the value of $L(w_1(k), w_2(k))$ or $\log_{10} L(w_1(k), w_2(k))$ (the latter might lead to better visualization, but you are open to choose whichever you deem more appropriate).

As a reference, your figure may look like the following figure. Note that this figure is not exactly reporting $\log_{10} L(w_1, w_2)$, but a related quantity.



Of course, your result won't be exactly the same, depending on the choice of initial weight and the range of learning rates. You are also free to choose a different colour scheme, etc.

1. [10pts] Pick a specific initial weights $w_1(0), w_2(0)$ and report it. Decide on the range of learning rates α_1 and α_2. Then generate the aforementioned type of figure for various choices of steps. For example, you can try $k \in \{10, 25, 50, 100, 200\}$. Describe how the figure changes as the iteration number k increases.
2. [5pts] How does the change of $w_1(0), w_2(0)$ affect your figure? Investigate several choices.
3. [3pts] For each group of figures you generated, discuss what ranges of learning rate leads to convergence and which values lead to divergence? You do not need to be exact, just a qualitative description based on the figures suffice.
4. [2pts] Zoom in close to the boundary of convergence and divergence and generate some of your figures again. Report your figures and explain your observations.
5. [5pts] Come up with a research question of yourself, state it, and investigate it, and report your results.

Question 5: Work Allocation [5 points]

This question is to make sure that if you are working with a partner, that you and your partner contributed equally to the assignment. If you are alone, just state that.

Please have each team member write down the approximate times that you worked on the assignment, and your contribution to the assignment.

```
# Example answer:
# I worked on the assignment on March 3rd afternoon, March 5th 12pm-2pm,
# and then March 6th in the evening. My partner and I had a meeting on
# March 3rd to read the entire assignment, and we did Question 1 together
# while screensharing. I worked out the math for Q2, and checked my
# partner's implementation in Q3. I also wrote the Q3 helper functions,
# and Q4(b).
```