Course Materials for GEN-AI

Northeastern University

These materials have been prepared and sourced for the course **GEN-AI** at Northeastern University. Every effort has been made to provide proper citations and credit for all referenced works.

If you believe any material has been inadequately cited or requires correction, please contact me at:

Instructor: Ramin Mohammadi
r.mohammadi@northeastern.edu

Thank you for your understanding and collaboration.

Homework - 1

Problem 1 (5 points)

Find the derivative of the softmax function.

Problem 2 (10 points)

Consider a model where the prior distribution over the parameters is a normal distribution with mean zero and variance σ^2, so that

$$Pr(\phi) = \prod_{j=1}^{J} \text{Norm}(0, \sigma_j^2; \phi_j),$$

where j indexes the model parameters. We now maximize $\prod_{i=1}^{n} Pr(y_i \mid x_i, \phi) Pr(\phi)$. Show that the associated loss function of this model is equivalent to L2 regularization.

Problem 3 (10 points)

Show that the weight decay parameter update with decay rate λ:

$$\phi \leftarrow (1 - \lambda)\phi - \alpha \frac{\partial L}{\partial \phi},$$

on the original loss function $L[\phi]$ is equivalent to a standard gradient update using L2 regularization so that the modified loss function $\tilde{L}[\phi]$ is:

$$\tilde{L}[\phi] = L[\phi] + \frac{\lambda}{2\alpha} \sum_{k} \phi_k^2,$$

where ϕ are the parameters, and α is the learning rate.

Problem 4 (8 points)

A network consists of three 1D convolutional layers. At each layer, a zero-padded convolution with kernel size three, stride one, and dilation one is applied. What size is the receptive field of the hidden units in the third layer?

Problem 5 (8 points)

A network consists of three 1D convolutional layers. At each layer, a zero-padded convolution with kernel size seven, stride one, and dilation one is applied. What size is the receptive field of hidden units in the third layer?

Problem 6 (8 points)

Consider a convolutional network with 1D input x. The first hidden layer H_1 is computed using a convolution with kernel size five, stride two, and a dilation rate of one. The second hidden layer H_2 is computed using a convolution with kernel size three, stride one, and a dilation rate of one. The third hidden layer H_3 is computed using a convolution with kernel size five, stride one, and a dilation rate of two. What are the receptive field sizes at each hidden layer?

Problem 7 (12 points)

A surface is guaranteed to be convex if the eigenvalues of the Hessian $H[\phi]$ are positive everywhere. In this case, the surface has a unique minimum, and optimization is easy. Find an algebraic expression for the Hessian matrix,

$$H[\phi] = \begin{bmatrix} \frac{\partial^2 L}{\partial \phi_0^2} & \frac{\partial^2 L}{\partial \phi_0 \partial \phi_1} \\ \frac{\partial^2 L}{\partial \phi_1 \partial \phi_0} & \frac{\partial^2 L}{\partial \phi_1^2} \end{bmatrix},$$

for the linear regression model. Prove that this function is convex by showing that the eigenvalues are always positive. This can be done by showing that both the trace and the determinant of the matrix are positive.

Linear Regression:

Consider applying gradient descent to the 1D linear regression model. The model $f(x_i, \phi)$ maps a scalar input x to a scalar output y and has parameters $\phi = (\phi_0, \phi_1)^T$, which represent the y-intercept and the slope:

$$y = f(x_i, \phi) = \phi_0 + \phi_1 x_i.$$

Given a dataset $\{(x_i, y_i)\}$ containing I input/output pairs, we choose the least squares loss function:

$$L[\phi] = \sum_{i=1}^{I} \left(f(x_i, \phi) - y_i\right)^2 = \sum_{i=1}^{I} \left(\phi_0 + \phi_1 x_i - y_i\right)^2,$$

Problem 8 (12 points)

The logistic regression model uses a linear function to assign an input to one of two classes $y \in \{0, 1\}$. For a 1D input and a 1D output, it has two parameters, ϕ_0 and ϕ_1, and is defined by:

$$Pr(y = 1|x) = \text{sig}(\phi_0 + \phi_1 x),$$

where sig is the logistic sigmoid function:

$$\text{sig}(z) = \frac{1}{1 + \exp(-z)}.$$

(i) Plot y against x for this model for different values of ϕ_0 and ϕ_1 and explain the qualitative meaning of each parameter.

(ii) What is a suitable loss function for this model?

(iii) Compute the derivatives of this loss function with respect to the parameters.

(iv) Generate ten data points from a normal distribution with mean -1 and standard deviation 1 and assign them the label $y = 0$. Generate another ten data points from a normal distribution with mean 1 and standard deviation 1 and assign these the label $y = 1$. Plot the loss as a heatmap in terms of the two parameters ϕ_0 and ϕ_1.

(v) Is this loss function convex? How could you prove this?

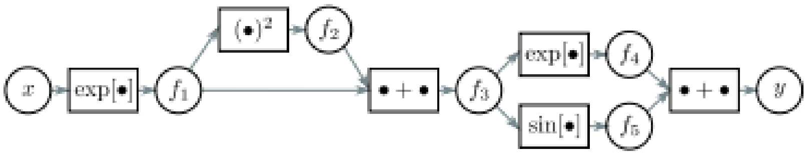

Figure 1: Image for question 9 and 10

Problem 9 (15 points)

This problem explores computing derivatives on general acyclic computational graphs. Consider the function:

$$y = \exp(\exp[x] + \exp[x]^2) + \sin(\exp[x] + \exp[x]^2).$$

We can break this down into a series of intermediate computations so that:

$$\begin{aligned} f_1 &= \exp[x], \\ f_2 &= f_1^2, \\ f_3 &= f_1 + f_2, \\ f_4 &= \exp[f_3], \\ f_5 &= \sin[f_3], \\ y &= f_4 + f_5. \end{aligned}$$

The associated computational graph is depicted in figure 1. Compute the derivative $\frac{\partial y}{\partial x}$ by reverse-mode differentiation. In other words, compute in order:

$$\frac{\partial y}{\partial f_5}, \frac{\partial y}{\partial f_4}, \frac{\partial y}{\partial f_3}, \frac{\partial y}{\partial f_2}, \frac{\partial y}{\partial f_1}, \text{ and } \frac{\partial y}{\partial x},$$

using the chain rule in each case to make use of the derivatives already computed.

Problem 10 (15 points)

For the same function as in problem 9, compute the derivative $\frac{\partial y}{\partial x}$ by forward-mode differentiation. In other words, compute in order:

$$\frac{\partial f_1}{\partial x}, \frac{\partial f_2}{\partial x}, \frac{\partial f_3}{\partial x}, \frac{\partial f_4}{\partial x}, \frac{\partial f_5}{\partial x}, \text{ and } \frac{\partial y}{\partial x},$$

using the chain rule in each case to make use of the derivatives already computed. Why do we not use forward-mode differentiation when we calculate the parameter gradients for deep networks?