CS 577 — Deep Learning — Midterm study guide

Throughout, we will write $\mathbf{x}^{(i)} \in \mathbb{R}^d$ for the i-th training sample and $y^{(i)}$ for the i-th training label. Let N denote the number of training samples.

1 Empirical risk minimization

- What is the "mean-square-error" (MSE) loss function used in regression? Write down the its formula $L(\hat{y}, y)$ where $\hat{y}$ is the model output and $y \in \mathbb{R}$ is the label.
- What is the "perceptron" loss used in binary classification? Write down its formula $L(z, y)$ where $z \in \mathbb{R}$ is the model output (before taking the sign) and $y \in \{\pm 1\}$ is the label.
- What is the "binary cross entropy" loss used in binary classification? Write down its formula $L(z, y)$ where z is the model output (before taking the sign) and $y \in \{\pm 1\}$ is the label.
    - Sketch the graph of $L(z, y)$ as a function of z for both $y = 1$ and $y = -1$.
    - Sketch the graph of $\frac{\partial L}{\partial z}(z, y)$ as a function of z for both $y = 1$ and $y = -1$.
- What is the $J(\boldsymbol{\theta})$ function (state your answer in terms of the MSE loss)?
- What is the $J_i(\boldsymbol{\theta})$ function (state your answer in terms of the MSE loss)?

2 Optmization

- What is gradient descent? Give the pseudocode of the algorithm in terms of $J(\boldsymbol{\theta})$ and step size $\eta > 0$.
- What is *stochastic gradient descent* (SGD)? Give the pseudocode for the case when the minibatch size is $= 1$ and the data is not shuffled.
- Perceptron algorithm
    - What is the perceptron algorithm for binary classification? Give the pseudocode.
    - Explain step-by-step how the perceptron algorithm is a type of SGD, i.e., what is the loss function, what is the minibatch size, what is the step size, how the pseudocode of SGD matches up with the pseudocode of the perceptorn algorithm

3 Regression

- Linear regression
    - Write down the mathematical expression for $J(\boldsymbol{\theta})$ when $\boldsymbol{\theta} = \begin{bmatrix} \mathbf{w} \\ b \end{bmatrix}$ where $\mathbf{w} \in \mathbb{R}^d$ is the "slope/weight" and b is the "intercept/bias".
    - What is the $\tilde{\mathbf{X}}$ trick from Lecture 02 for absorbing the bias?
    - What is the formula for the gradient $\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$? Use $\tilde{\mathbf{X}}$. Be able to fill in the following code:

```python
# assume that
# X.shape = (N,2)
# y.shape = (N,)
# w.shape = (2,)
# b is a float

theta = np.concatenate([w, np.array([b])])
Xtilde =
grad_theta =
```

- Feature map
  - What is the feature map $\phi : \mathbb{R}^d \to \mathbb{R}^D$ for all polynomials up to some given `degree` when $d = 1$? You should be able to answer this in code:

```python
def polynomial_feature_map_(x,degree):
  # Note: x is a number, i.e., an array with shape=()

  # COMPLETE THE LINE BELOW

  return

def polynomial_feature_map(x_array,degree):
  # Note: This is not vectorized. That is okay for this problem.
  return np.array([polynomial_feature_map_(x_array[i],degree) for i in range(len(x_array))])
```

  - Be able to sketch by hand the following plot

```python
x = np.linspace(-1, 2, 100)
degree = 2
w = np.array([1, -2, 1])
X_tilde = polynomial_feature_map(x, degree)
y = X_tilde @ w
plt.plot(x, y)
```

4 Probability

- Regression (where $y^{(i)} \in \mathbb{R}$)
  - Be able to explain how to go from the conditional distribution

$$p(y^{(i)} \mid \mathbf{x}^{(i)}; \boldsymbol{\theta}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2}\frac{(y^{(i)} - f(\mathbf{x}^{(i)}; \boldsymbol{\theta}))^2}{\sigma^2}\right) \tag{1}$$

to the mean squared error loss

$$L(\hat{y}^{(i)}, y^{(i)}) = (\hat{y}^{(i)} - y^{(i)})^2 \quad \text{where} \quad \hat{y}^{(i)} = f(\mathbf{x}^{(i)}; \boldsymbol{\theta}). \tag{2}$$

  - How to generate $y^{(i)}$ from the conditional distribution (1) given $\mathbf{x}^{(i)}$? You should be able to answer this question in code:

```python
w = np.array([1.0,2.0])
b = 3.0
x = np.random.randn(N,2)
sigma = 0.5

# COMPLETE THE UNFINISHED LINE

eps =

y = x@w + b + eps
```

- Classification (where $y^{(i)} \in \{1, \ldots, K\}$)
  - For $\mathbf{z} = \begin{bmatrix} z_1 & \cdots & z_K \end{bmatrix} \in \mathbb{R}^K$, how is the softmax of $\mathbf{z}$ defined? What is the formula for $\text{softmax}(\mathbf{z})_y$ where $y \in \{1, \ldots, K\}$?
  - How is the cross entropy loss $L(\mathbf{z}^{(i)}, y^{(i)})$ defined in terms of the softmax and $\mathbf{z}^{(i)} = f(\mathbf{x}^{(i)}; \boldsymbol{\theta})$?
  - What is the formula for $\frac{\partial L}{\partial \mathbf{z}}(\mathbf{z}, y)$ where L is the cross entropy loss?
  - Consider the following code block:

```
loss_der = np.zeros((n_samples, n_classes))    # (n_samples, n_classes) = (N,K)
h,z = forward(X,theta)
assert(z.shape == (n_samples, n_classes))    # model output
# Question: how to calculate loss_der, the loss derivative, in a vectorized way?
# On the exam, you will be asked to fill in code

expz = np.exp(z)

# COMPLETE THE UNFINISHED LINE

p =

y_one_hot =

loss_der =

# HINT: see lec05-in-class-exercise
```

5 Tensor manipulation

- How can I vectorize the following using `np.matmul`?

```
dJdW2 = np.zeros((10,3))    # 10 = number of neurons, 3 = number of classes
assert(relu_h.shape == (150,10)) # 150 = number of training samples
assert(loss_der.shape == (150,3))
for i in range(n):
    dJdW2 += np.outer(relu_h[i,:], loss_der[i,:])

dJdW2 /= n

# COMPLETE THE FOLLOWING LINE WHICH VECTORIZES THE ABOVE OPERATION

dJdW2 = np.matmul(
```

- Coding completion problems related to homework 3 similar to the above.

6 Neural networks

- What is the relu activation function (denote it by g)? Plot g and its derivative g' over $[-2, 2]$.
- Plot over $x \in [-1, 3]$ the function

$$f(x; \boldsymbol{\theta}) = g(m_1 x + b_1) - g(m_2 x + b_2) \quad \text{where} \quad \boldsymbol{\theta} = [m_1, b_1, m_2, b_2] = [1, -1, 1, -2]$$

  You will be given a hint like: "it should be a piecewise linear function with 3 linear pieces".

- Calculate the derivatives

$$\frac{\partial J_i}{\partial m_1}(\boldsymbol{\theta}) \qquad \frac{\partial J_i}{\partial b_1}(\boldsymbol{\theta}) \qquad \frac{\partial J_i}{\partial m_2}(\boldsymbol{\theta}) \qquad \frac{\partial J_i}{\partial b_1}(\boldsymbol{\theta})$$

  where

$$J_i(\boldsymbol{\theta}) = (y^{(i)} - f(x^{(i)}; \boldsymbol{\theta}))^2$$

  for some $x^{(i)} \in \mathbb{R}$ and $y^{(i)} \in \mathbb{R}$.