Midterm Review

CS 4277: Deep Learning

1. According to Tom Mitchell, machine learning is the study of algorithms that

   - improve their performance `P`
   - at some task `T`
   - with experience `E`.

   A well-defined learning task is given by `<P, T, E>`.

   Formulate the following problems according to Tom Mitchell's machine learning problem specification (see Machine Learning Slides) and the specification our textbook. For each of the following problems specify:

   - The task `T`,
   - The performance measure `P`,
   - The experience `E`,
   - The target function $f : \mathcal{X} \to \mathcal{Y}$, that is,
     - the input space $\mathcal{X}$, and
     - the output space $\mathcal{Y}$.

   Remember that a function maps a domain to a co-domain, and these domains are sets.

2. Medical diagnosis: A patient walks in with a medical history and some symptoms, and you want to identify the problem.

   Solution:

   - Task, `T`: diagnose problem
   - Performance, `P`: diagnosis is correct or incorrect
   - Experience, `E`: $< medical - history, symptoms >$
   - Target function $f : \mathcal{X} \to \mathcal{Y}$:
     - $\mathcal{X} = \{\vec{x} | x_1 \in \{family - history - heart - disease\}, x_2 \in \mathbb{R} = cholesterol - level\}$ and other such features
     - $\mathcal{Y} = \{disease_1, disease_2, ..., disease_n\}$

3. For a single neuron with a two-dimensional input $\boldsymbol{x}$ and single scalar output y, formulate the preactivation value using a linear algebra operation. Don't forget to account for the bias.

Solution:

$$\boldsymbol{x}^T\boldsymbol{\theta} = \begin{bmatrix} 1 & x_1 & x_2 \end{bmatrix} \begin{bmatrix} \theta_0 \\ \theta_2 \\ \theta_2 \end{bmatrix} = \sum_{i=1}^{D} x_i\theta_i \tag{1}$$

4. Write a mathematical definition of the ReLU activation function.

Solution:

$$a(z) = RelU(z) = \begin{cases} 0, & z < 0 \\ z, & z \geq 0 \end{cases} \tag{2}$$

5. The squared error loss function for a regression model with a single scalar input and single scalar output is:

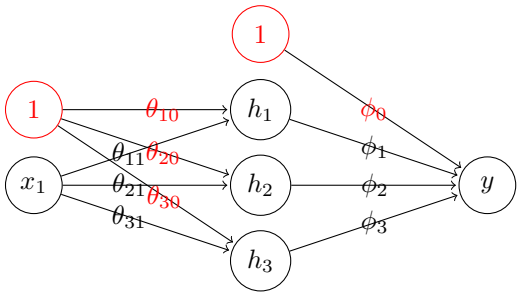
$$L(\boldsymbol{\phi}) = \sum_{i=1}^{I} (\phi_0 + \phi_1 x_i - y_i)^2 \tag{3}$$

Write the squared error loss function for a regression model with a two-dimensional vector input, $\boldsymbol{x}$, and a single scalar output y.

Solution:

$$L(\boldsymbol{\phi}) = \sum_{i=1}^{I} (\phi_0 + \phi_1 x_{i1} + \phi_2 x_{i2} - y_i)^2 \tag{4}$$

6. Given the following shallow neural network:



Write a single equation for the network $y = f(x, \vec{\phi}$ with parameters $\boldsymbol{\phi} = \{\boldsymbol{\theta}, \boldsymbol{\phi}\}$.

Solution:

$$y = f(x, \vec{\phi}) = \phi_0 + \phi_1 a(\theta_{10} + \theta_{11}x) + \phi_2 a(\theta_{20} + \theta_{21}x) + \phi_3 a(\theta_{30} + \theta_{31}x) \tag{5}$$

7. State the universal approximation theorem.

Solution: For any function, there exits a shallow network with enough hidden units to approximate the function to any precision.

8. A shallow network with $D > 2$ hidden units can create up to how many linear regions?

Solution:

$$D + 1$$

9. A deep network with K layers of $D > 2$ hidden units can create up to how many linear regions?

Solution:

$$(D + 1)^K$$

10. Explain the *depth efficiency* of deep networks vs shallow networks.

Solution: Some functions require exponentially more hidden units in a shallow network than an equivalent deep network.

11. Explain the i.i.d. assumption.

Solution: We assume training data points are drawn independently and are identically-distributed.

12. What is a loss function?

Solution: A loss function returns a single number that represents the mismatch between the model's output $\hat{y}$ and the ground truth y.

13. How do machine learning algorithms use loss functions?

Solution: A machine learning algorithm minimizes the loss function, which represents the best possible mapping from training inputs to outputs.

14. Write a concise general mathematical formulation of the goal of a machine learning algorithm that relates the model's parameters to the loss function.

Solution:

$$\hat{\boldsymbol{\phi}} = \underset{\boldsymbol{\phi}}{\operatorname{argmin}}(L(\boldsymbol{\phi}))$$

15. Write the general 2-step gradient descent algorithm.

Solution: 1. Compute the derivatives of the loss with respect to the parameters.

$$\nabla L = \frac{\partial L}{\partial \boldsymbol{\phi}} = \begin{bmatrix} \frac{\partial L}{\partial \boldsymbol{\phi}_0} \\ \frac{\partial L}{\partial \boldsymbol{\phi}_1} \\ \vdots \\ \frac{\partial L}{\partial \boldsymbol{\phi}_N} \end{bmatrix}$$

2. Update the parameters according to the update rule:

$$\boldsymbol{\phi} \leftarrow \boldsymbol{\phi} - \alpha \frac{\partial L}{\partial \boldsymbol{\phi}}$$

where α is a positive scalar value called the *learning rate* that controls how large parameter updates are in each training step.

16. What are local minima? What is the global minimum?

Solution: The global minimum is the point at which the function's value is minimal. Local minima are points at which the function's value is lower than it's neighbors, but not as low as the global minimum.

17. What if the learning rate is set too high?

Solution: Won't converge to minimum because updates will "jump over" the minimum.

18. Is full-batch gradient descent guaranteed to find the global minimum? Why or why not?

Solution: No. If the loss function has local minima in addition to the global minimum, full-batch gradient descent may converge to a local minimum. If there are many local minima or the parameters happen to be initialized nearer to a local minumum than the global minimum, the chances of converging to a local minimum increase.

19. Describe stochastic gradient descent and how it improves on full-batch gradient descent.

Solution: At each iteration, the algorithm chooses a random subset of the training data and computes the gradient from these examples alone. This subset is known as a minibatch or batch for short. Batches are drawn without replacement, and iterations continue with new batches until all training data points have been used – an epoch.

Beacause the gradient is computed on a subset of the data, the direction of parameter updates might not be the steepest descent, or may actually be a "climb" with respect to the full data set. This has the effect of "climbing out" of regions of local minima. SGD is not guaranteed to find the global minimum, but in practice SGD gives good results.

20. Describe SGD modified with momentum and the purpose of momentum.

Solution: We update the parameters with a weighted combination of the gradient computed from the current batch and the direction moved in the previous step:

$$\boldsymbol{m}_{t+1} \leftarrow \beta \boldsymbol{m}_t + (1-\beta) \sum_{i \in \mathcal{B}} \frac{\partial \ell_i(\boldsymbol{\phi})}{\partial \boldsymbol{\phi}}$$

$$\boldsymbol{\phi_{t+1}} \leftarrow \boldsymbol{\phi}_t - \alpha \boldsymbol{m}_{t+1}$$

The overall effect is a smoother trajectory and reduced oscillatory behavior in valleys of the loss function surface.

21. Describe Adam and its purpose.

Solution: Adam stands for *adaptive moment estimation*. The purpose of Adam is to normalize the gradients so that we move a fixed distance (governed by the learning rate) in each direction. Without this normalization, updates are greater for parameters with the steepest gradients. Because the magnitude of updates is different for each parameter it is difficult to choose a learning rate that that makes good progress in each direction and is stable.

22. What are the components of the Hessian matrix?

Solution: The second derivatives of the loss function with respect to the model parameters.

23. What is useful about the Hessian matrix?

Solution: If the Hessian matrix is positive definite, that is, all its eigenvalues are positive for any parameter values, then the loss function is convex, meaning it has a single global minimum with no local minima. In this case gradient descent is guaranteed to converge to the global minimum given suitable learning rate.

24. What is computed in the forward pass of the backpropagation algorithm?

Solution: The input is *fed forward* to compute all intermediate values: all the preactivations, f_k, and activations, h_k, the output, and the loss, $y - \hat{y}$, are computed and stored.

25. What is computed in the backward pass of the backpropagation algorithm?

Solution: The derivatives of the loss ℓ_i with respect to the intermediate values computed in the forward pass, then the derivatives of the loss ℓ_i with respect to the parameters β_k and ω_k.

26. Describe the efficiency of the backpropagation algorithm with respect to time and memory.

Solution: Backpropagation is time efficient but requires a great deal of storage to store intermediate values.

27. How are the weights of a network typically initialized?

Solution: With values drawn from a Guassian with $\mu = 0$ and some σ^2.

28. Describe the vanishing gradients problem and what initialization mistake leads to it.

Solution: If the variance σ^2 of the Gaussian used to initialize the parameters of the network is too small, then the weights may become very low, underflowing floating-point representation capability.

29. Describe the exploding gradients problem and what initialization mistake leads to it.

Solution: If the variance σ^2 of the Gaussian used to initialize the parameters of the network is too large, then the weights may become very large and training will become unstable.

30. What initialization method avoids the vanishing gradients and exploding gradients problem?

Solution: *He initialization*: initialize a given layer with:

$$\sigma_{\Omega}^2 = \frac{2}{D_h}$$

Where D_h is the dimension of the layer feeding the layer for which we're setting the weights.