Return to Edit Assessment

Name: _____
Score: _____ / _____

Worksheet 03 - networks and minimization

Part 1

Please state the names of all the students you worked with on this assignment:

Answer Point Value: 0.0 points
Model Short Answer: --------

Part 2: Neural networks

Consider a densely connected two-layer neural network with one input and one output node and 10 nodes in the hidden layer. The network parameters Θ are defined by the weights of the input to the hidden and the hidden to the output layer, as well as the biases of the hidden and output neurons. The hidden and output neurons have sigmoid activation functions $f(x) = (1 + e^{-x})^{-1}$.

For this neutral network, there is a setting of parameters Θ, such that the network approximates any given smooth function $y(x)$ in the sense

$$|\hat{y}(x;\Theta) - y(x)| < \epsilon$$

for all $x \in \mathbb{R}$ for a given $\epsilon > 0$.

- True
- False

Answer Point Value: 1.0 points
Answer Key: False

In a two-layer neural network (input, hidden, output), with sigmoid activation functions, how many hidden neurons are at least needed to approximate the function

$$y : X = [-2d, 2d] \to \mathbb{R},\ y(x) = h(x + d) - h(x - d)$$

with the Heavyside step function

$$h(x) = \begin{cases} 0 & x < 0 \\ 1 & x \geq 0 \end{cases}$$

in the sense $\|\hat{y}(\cdot;\Theta) - y(\cdot)\|_{L^2} < 10^{-5}$, where $\|f\|_{L^2} := (\int_X f^2)^{\frac{1}{2}}$?

- A. 1
- B. 2
- C. 4
- D. ∞

Answer Point Value: 1.0 points

Answer Key: B

We are given a densely connected neural network called **autoencoder** with 100 input and 100 output neurons. In between we have hidden layers with (30, 10, 2, 10, 30) neurons. The network parameters Θ are defined by the weights and biases of all layers. Compute the number of free (learnable) parameters and state, assuming that we store them in single precision floating point numbers (float32), how much memory we need.

- A. $1,000 - 1,500$ **Byte**
- B. $26,000 - 26,999$ **Byte**
- C. $27,000 - 27,999$ **Byte**
- D. **More than** $1,000,000$ **Byte**

Answer Point Value: 1.0 points

Answer Key: C

Part 3: Minimization

Which of these packages does not offer neural network optimization with graphical processor unit (GPU) acceleration?

- A. **PyTorch**
- B. **scikit-learn**
- C. **Tensorflow**
- D. **Keras**

Answer Point Value: 1.0 points

Answer Key: B

We want to minimize the functions

$$\begin{aligned} C_1(\Theta) &= \tfrac{1}{2}\theta_1^2 + \tfrac{1}{2}\theta_2^2 \\ C_2(\Theta) &= \tfrac{1}{2}\theta_1^2 + \tfrac{1}{2000}\theta_2^2 \end{aligned}$$

using either simple gradient descent or the Newton method starting from an initial point $\Theta_0 = (1,1)^\top$.

For gradient descent, assume we use the same learning rate for both functions and we choose this learning rate such that the algorithm converges asymptotically. Suppose that, if we minimize C_1 with gradient descent, we need $1,000$ iterations to reach the minimium $\Theta = (0,0)^\top$ with a small error tolerance ϵ.

How many steps (order of magnitude) do you expect gradient descent and the Newton method to take for minimizing C_2 to within the same error tolerance ϵ?

- A. $\mathcal{O}(1)$ **for Newton and** $\mathcal{O}(1)$ **for gradient descent.**
- B. $\mathcal{O}(1)$ **for Newton and** $\mathcal{O}(10^3)$ **for gradient descent.**
- C. $\mathcal{O}(1)$ **for Newton and** $\mathcal{O}(10^6)$ **for gradient descent.**
- D. $\mathcal{O}(10^3)$ **for Newton and** $\mathcal{O}(10^3)$ **for gradient descent.**
- E. $\mathcal{O}(10^3)$ **for Newton and** $\mathcal{O}(10^6)$ **for gradient descent.**
- F. $\mathcal{O}(10^3)$ **for Newton and** $\mathcal{O}(1)$ **for gradient descent.**

Answer Point Value: 1.0 points

Answer Key: C

Consider the optimization of the quadratic function

$$C(\Theta) = \Theta^\top \mathbf{A} \Theta$$

where $\Theta \in \mathbb{R}^N$ is the parameter vector and $\mathbf{A} \in \mathbb{R}^{N \times N}$ is a dense matrix. We consider minimizing this function using Newton's method or gradient descent from an initial point Θ_0. How much slower is a single Newton step than a single gradient descent step (order of magnitude)?

- A. $\mathcal{O}\left(N^{-1}\right)$
- B. $\mathcal{O}\left(1\right)$
- C. $\mathcal{O}\left(N\right)$
- D. $\mathcal{O}\left(N^2\right)$
- E. $\mathcal{O}\left(N^3\right)$

Answer Point Value: 1.0 points

Answer Key: D

We want to train a deep neural network by minimizing a given loss function. We assume that the global minimum of the loss function is significantly lower than local minima. State which minimizer you recommend for reliably finding this global minimum from an arbitrary starting point and be prepared to argue your choice.

- A. **Gradient descent**

B. **Newton method**

C. **Stochastic gradient descent**

Answer Point Value: 1.0 points

Answer Key: C

We have built a neural network with the loss function $C(\Theta)$. We want to use stochastic gradient descent with batchsize B and learning rate η to train the network. Which of these strategies is the best to learn a model that makes reliable predictions?

A. **Set** $B = N(\text{data size})$ **and** $\eta = 10^2$ **. Find parameters** Θ **by minimizing** $C(\Theta)$**.**

B. **Divide the data into training and validation set** $\left(\mathbf{X}^{\text{train}}, \mathbf{X}^{\text{val}}\right)$**. For different combinations of** (B, η)**, optimize**

$$\Theta^* = \arg\min_{\Theta} C\left(\Theta, \mathbf{X}^{\text{train}}\right)$$

and call the resulting loss $C_{B,\eta}^{\text{train}} = C\left(\Theta^*, \mathbf{X}^{\text{train}}\right)$**. Use the solution with**

$$\Theta^\dagger = \arg\min_{B,\eta} C_{B,\eta}^{\text{train}}$$

C. **Same as (B), but use the solution**

$$\Theta^\dagger = \arg\min_{B,\eta} C_{B,\eta}^{\text{val}}$$

where $C_{B,\eta}^{\text{val}} = C\left(\Theta^*, \mathbf{X}^{\text{val}}\right)$**.**

D. **Choose all parameters** (Θ, B, η) **by minimizing over the joint space defined by** (Θ, B, η) **using all data.**

Answer Point Value: 1.0 points

Answer Key: C

We consider fitting a function $y(x) : \mathbb{R}^{10} \rightarrow \mathbb{R}$ by training a neural network. We consider a 10-layer dense network, but want to find the best number of neurons in each layer. We want to use one nonlinear activation function, but are not sure which one. We will use the Adam optimizer to train the neural network.

How many dimensions does the hyperparameter space have?

- A. 8
- B. 10
- C. 14 **correct answer**
- D. 16
- E. 105