1. An e-commerce company in Pakistan wants to build a system that can classify product reviews as positive, negative, or neutral. They have collected a dataset of 5000 reviews with their corresponding sentiment labels (positive, negative, neutral). The reviews contain text data, and the company wants to use supervised learning to train a model that can predict the sentiment of new, unseen reviews.
   The dataset contains the following features:
   A) Review text
   B) Product category (electronics, clothing, etc.)
   C) User rating (1-5 stars)
   D) Review length (number of words)

   What type of supervised learning problem is this scenario an example of?

   A) Regression
   B) Classification
   C) Clustering
   D) Dimensionality Reduction

   **Answer: B)**

**2.** Which of the following is an example of unsupervised learning?
   A) Image classification using convolutional neural networks
   B) Object detection using YOLO algorithm
   C) Clustering customers based on their buying behavior
   D) Predicting house prices using linear regression

   **Answer: B)**

3. What is the primary difference between supervised and unsupervised learning?
   A) Supervised learning uses neural networks, while unsupervised learning uses decision trees
   B) Supervised learning uses labeled data, while unsupervised learning uses unlabeled data
   C) Supervised learning is used for regression tasks, while unsupervised learning is used for classification tasks
   D) Supervised learning is used for clustering tasks, while unsupervised learning is used for dimensionality reduction

   **Answer: B**

4. What is the activation pattern for each of the center regions in figure given? In other words, which hidden units are active (pass the input) and which are inactive (clip the input) for center region?

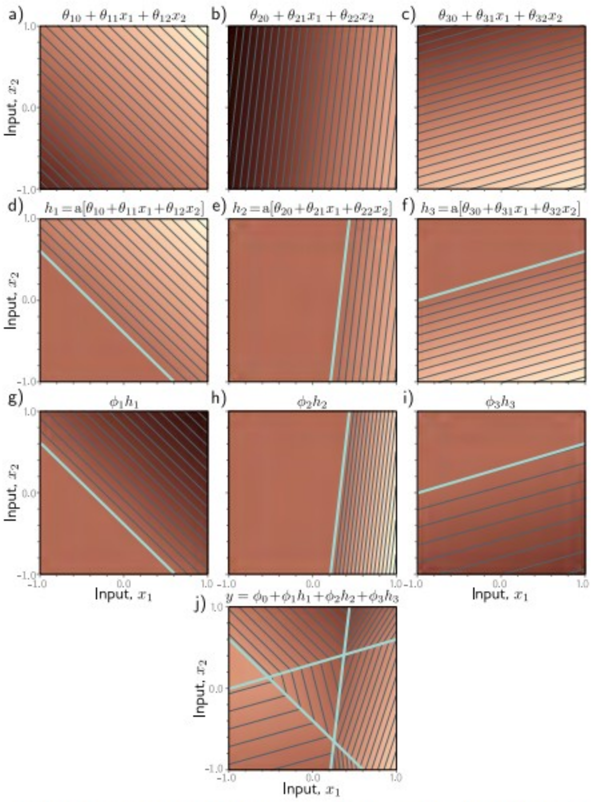
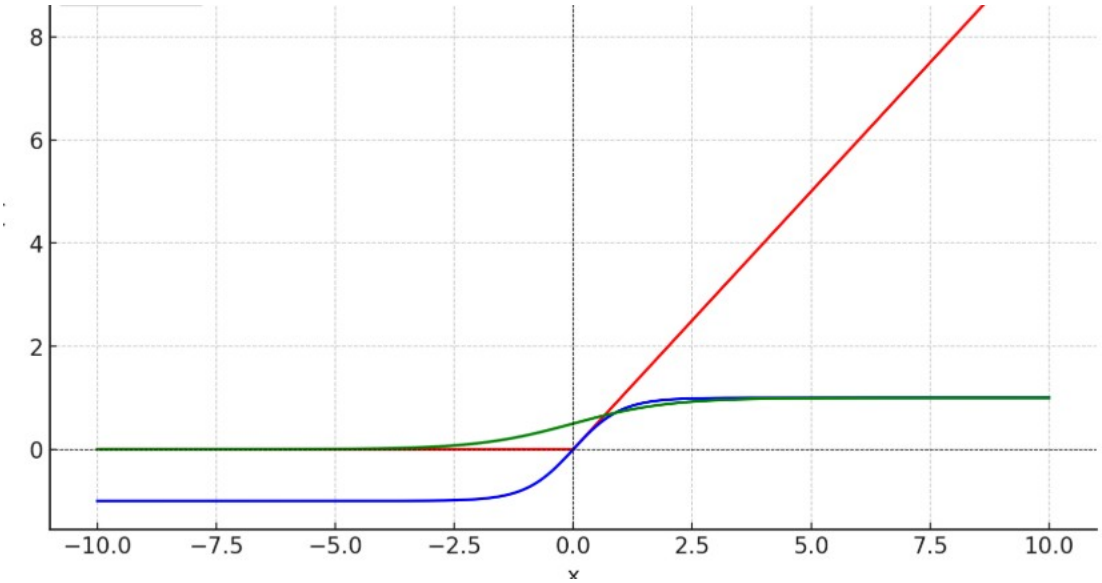


**Figure 3.8** Processing in network with two inputs $\mathbf{x}=[x_1,x_2]^T$, three hidden units $h_1,h_2,h_3$, and one output $y$. a–c) The input to each hidden unit is a linear function of the two inputs, which corresponds to an oriented plane. Brightness indicates function output. For example, in panel (a), the brightness represents $\theta_{10}+\theta_{11}x_1+\theta_{12}x_2$. Thin lines are contours. d–f) Each plane is clipped by the ReLU activation function (cyan lines are equivalent to "joints" in figures 3.3d–f). g-i) The clipped planes are then weighted, and j) summed together with an offset that determines the overall height of the surface. The result is a continuous surface made up of convex piecewise linear polygonal regions.

A) Center region (active, inactive, active)
B) Center region (inactive, inactive, active)
C) Center region (inactive, active, active)
D) Center region (active, active, active)

**Answer: A**

**5.**



Here is a graph showing the activation functions: ReLU, Tanh, and Sigmoid.

Which line in the graph represents the Tanh function?

A) Red Line
B) Green Line
C) Blue Line
D) None of the above

**Answer: B**

6. Which of the following best describes the Universal Approximation Theorem?

A) A neural network with one hidden layer can approximate any discrete function given enough neurons in the hidden layer.

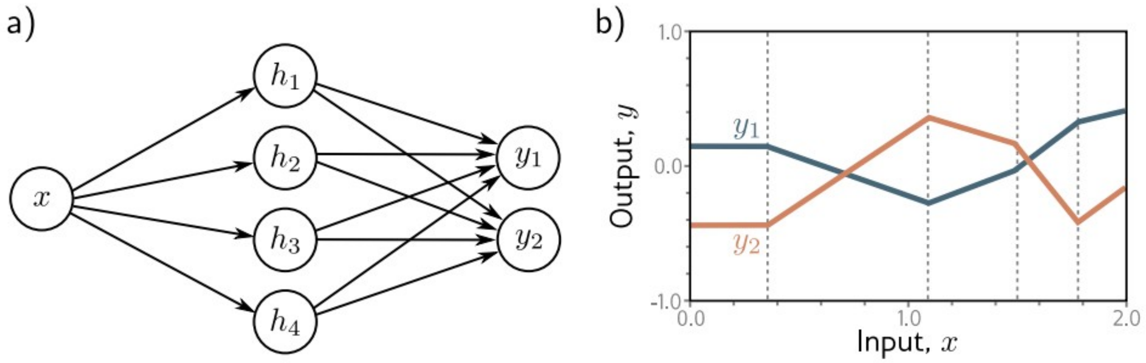
B) A neural network with two or more hidden layers can learn any linear transformation regardless of the activation function used.

C) A neural network with a single hidden layer containing a finite number of neurons can approximate any continuous function on a compact subset, given a suitable non-linear activation function.

D) A neural network can only achieve universal approximation if the activation function is linear, like ReLU.

**Answer: C**

7. Given the below figure



What is the reason why the "joints" of the piecewise linear functions  and  are constrained to be in the same places in the network shown?

A) The network uses a shared activation function for both outputs.

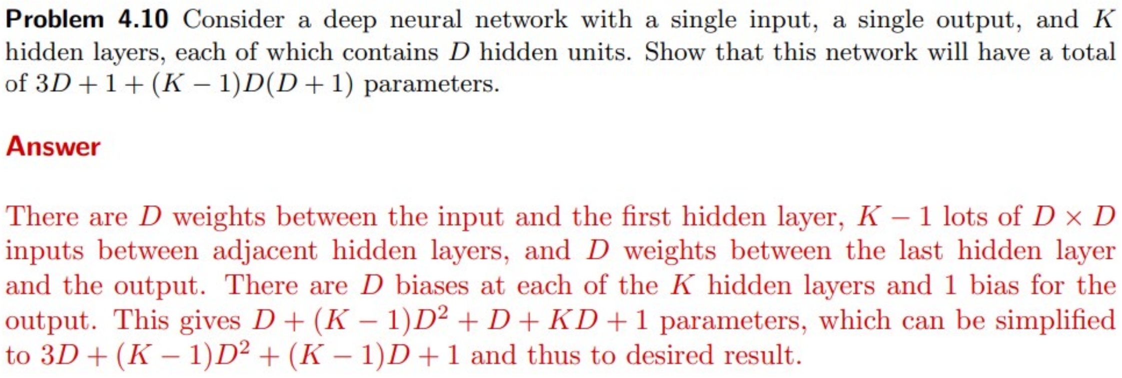
B) Both outputs share the same set of input data points.

C) The network shares the same hidden units for both output calculations.

D) The network employs a linear transformation for each hidden layer.

**Answer: C**

8. **Give this as fill in the blanks example:**

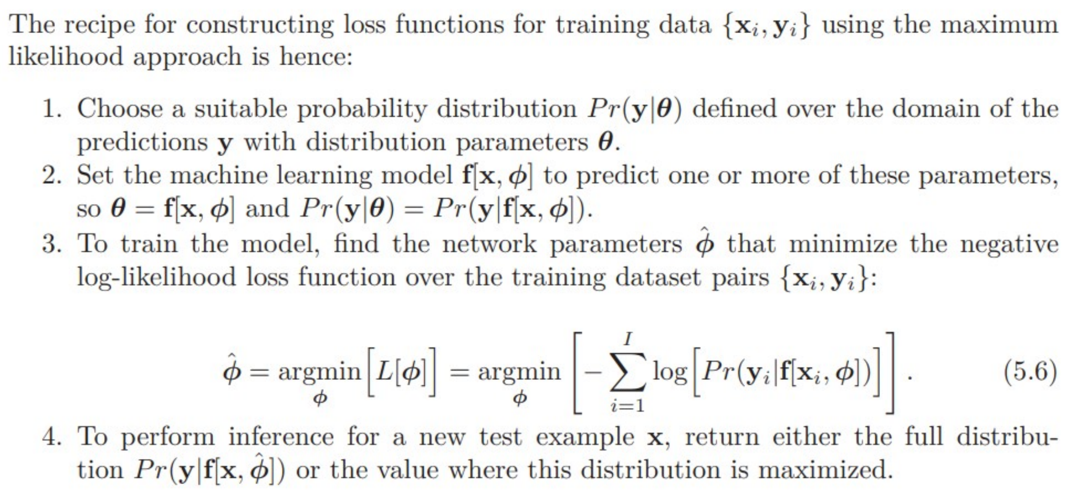


**Problem 4.10** Consider a deep neural network with a single input, a single output, and $K$ hidden layers, each of which contains $D$ hidden units. Show that this network will have a total of $3D + 1 + (K - 1)D(D + 1)$ parameters.

**Answer**

There are $D$ weights between the input and the first hidden layer, $K - 1$ lots of $D \times D$ inputs between adjacent hidden layers, and $D$ weights between the last hidden layer and the output. There are $D$ biases at each of the $K$ hidden layers and 1 bias for the output. This gives $D + (K - 1)D^2 + D + KD + 1$ parameters, which can be simplified to $3D + (K - 1)D^2 + (K - 1)D + 1$ and thus to desired result.

9. **Answer the questions based on the image below**

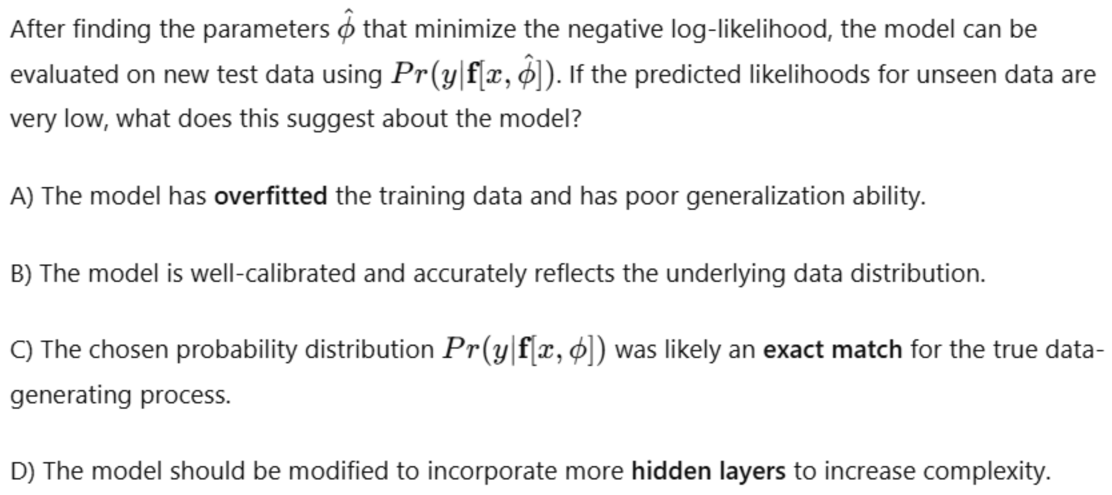


The recipe for constructing loss functions for training data $\{\mathbf{x}_i, \mathbf{y}_i\}$ using the maximum likelihood approach is hence:

1. Choose a suitable probability distribution $Pr(\mathbf{y}|\boldsymbol{\theta})$ defined over the domain of the predictions $\mathbf{y}$ with distribution parameters $\boldsymbol{\theta}$.
2. Set the machine learning model $\mathbf{f}[\mathbf{x}, \boldsymbol{\phi}]$ to predict one or more of these parameters, so $\boldsymbol{\theta} = \mathbf{f}[\mathbf{x}, \boldsymbol{\phi}]$ and $Pr(\mathbf{y}|\boldsymbol{\theta}) = Pr(\mathbf{y}|\mathbf{f}[\mathbf{x}, \boldsymbol{\phi}])$.
3. To train the model, find the network parameters $\hat{\boldsymbol{\phi}}$ that minimize the negative log-likelihood loss function over the training dataset pairs $\{\mathbf{x}_i, \mathbf{y}_i\}$:

$$\hat{\boldsymbol{\phi}} = \underset{\boldsymbol{\phi}}{\operatorname{argmin}}\Big[L[\boldsymbol{\phi}]\Big] = \underset{\boldsymbol{\phi}}{\operatorname{argmin}}\left[-\sum_{i=1}^{I}\log\Big[Pr(\mathbf{y}_i|\mathbf{f}[\mathbf{x}_i, \boldsymbol{\phi}])\Big]\right]. \qquad (5.6)$$

4. To perform inference for a new test example $\mathbf{x}$, return either the full distribution $Pr(\mathbf{y}|\mathbf{f}[\mathbf{x}, \hat{\boldsymbol{\phi}}])$ or the value where this distribution is maximized.



After finding the parameters $\hat{\phi}$ that minimize the negative log-likelihood, the model can be evaluated on new test data using $Pr(y|\mathbf{f}[x, \hat{\phi}])$. If the predicted likelihoods for unseen data are very low, what does this suggest about the model?

A) The model has **overfitted** the training data and has poor generalization ability.

B) The model is well-calibrated and accurately reflects the underlying data distribution.

C) The chosen probability distribution $Pr(y|\mathbf{f}[x, \phi])$ was likely an **exact match** for the true data-generating process.

D) The model should be modified to incorporate more **hidden layers** to increase complexity.

**Answer A**

10. In the context of this approach, which distribution would be most appropriate if the target variable follows a binary outcome (0 or 1)?

A) Gaussian (Normal) Distribution

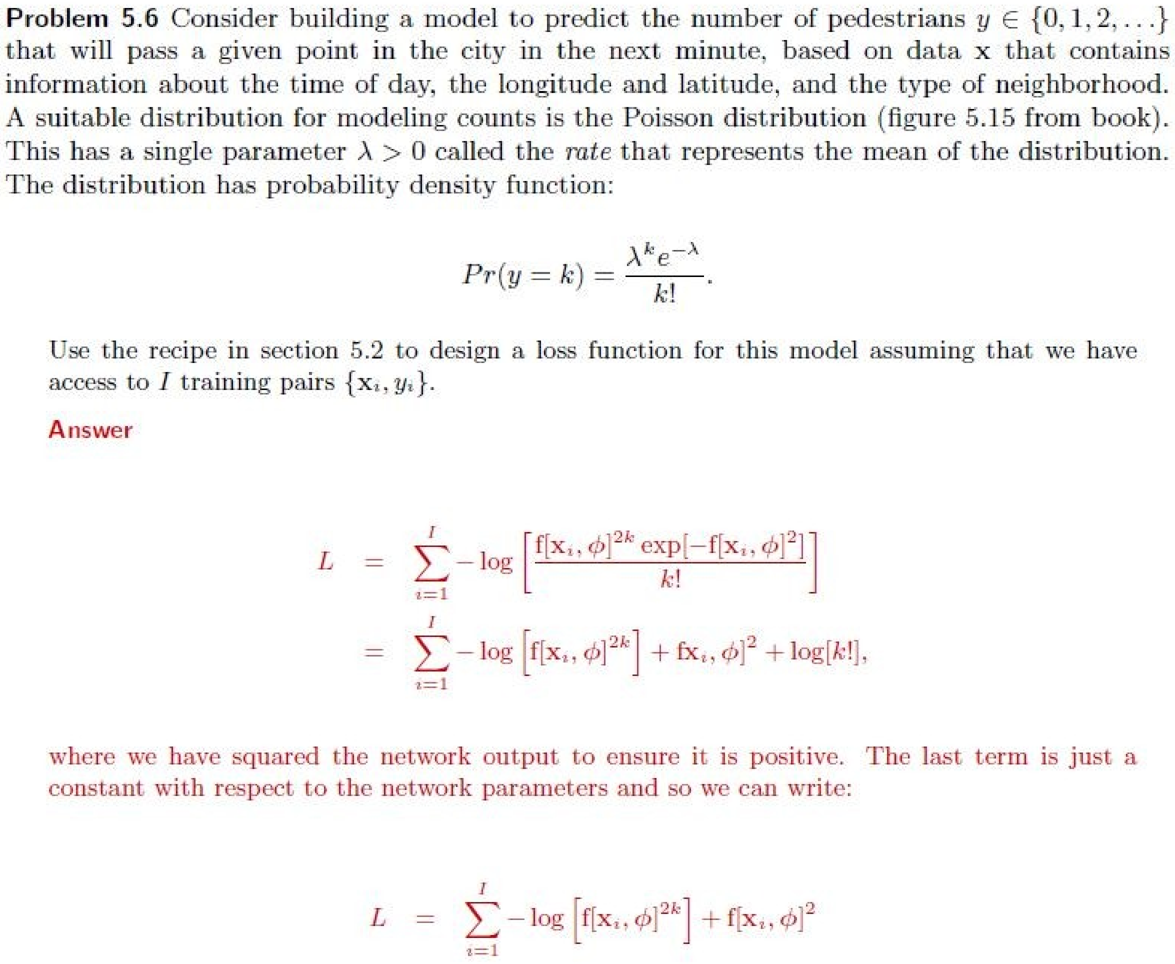
B) Poisson Distribution

C) Bernoulli Distribution

D) Exponential Distribution

Answer: C

11.



**Problem 5.6** Consider building a model to predict the number of pedestrians $y \in \{0, 1, 2, \ldots\}$ that will pass a given point in the city in the next minute, based on data $\mathbf{x}$ that contains information about the time of day, the longitude and latitude, and the type of neighborhood. A suitable distribution for modeling counts is the Poisson distribution (figure 5.15 from book). This has a single parameter $\lambda > 0$ called the *rate* that represents the mean of the distribution. The distribution has probability density function:

$$Pr(y = k) = \frac{\lambda^k e^{-\lambda}}{k!}.$$

Use the recipe in section 5.2 to design a loss function for this model assuming that we have access to $I$ training pairs $\{\mathbf{x}_i, y_i\}$.

**Answer**

$$\begin{aligned} L &= \sum_{i=1}^{I} -\log\left[\frac{\mathrm{f}[\mathbf{x}_i, \phi]^{2k} \exp[-\mathrm{f}[\mathbf{x}_i, \phi]^2]}{k!}\right] \\ &= \sum_{i=1}^{I} -\log\left[\mathrm{f}[\mathbf{x}_i, \phi]^{2k}\right] + \mathrm{f}\mathbf{x}_i, \phi]^2 + \log[k!], \end{aligned}$$

where we have squared the network output to ensure it is positive. The last term is just a constant with respect to the network parameters and so we can write:

$$L = \sum_{i=1}^{I} -\log\left[\mathrm{f}[\mathbf{x}_i, \phi]^{2k}\right] + \mathrm{f}[\mathbf{x}_i, \phi]^2$$

12. Consider a neural network model designed to perform multiclass classification over classes. The output layer of the model uses a **softmax activation function** to predict the probabilities for each class. Letrepresent the predicted probability for class , and let be the one-hot encoded true label for class (1 if the true class is , 0 otherwise). The **cross-entropy loss function** for a single data point is given by:

$$L = -\sum_{i=1}^{C} y_i \log(\hat{y}_i)$$

Assume you have a dataset with 4 classes, and the network'spredictions for a single instance are as follows:

The true label for this instance is the second class, meaning . What is the cross-entropy loss for this instance?

A) 0.155

B) 0.357

C) 0.500

D) 0.714

Answer **B (if anyone is using log base 10 and their answer is 0.155, this has been marked correct as well).**

The **squared error loss** (also known as mean squared error, **MSE**) is often used to measure the performance of regression models. For a single data point, the squared error loss is defined as:

where:

- is the true value,

- is the predicted value.

For a dataset with samples, the mean squared error is given by:

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

You are given the following true and predicted values for a regression model:

True values

Predicted values

Calculate the mean squared error (MSE) for these predictions.

- A) 0.055

- B) 0.085
- C) 0.205
- D) 0.312

Solution

14.

Gradient Descent is an optimization algorithm used to minimize the loss function in machine learning models by iteratively updating the model parameters. In a simple linear regression model, the parameters  (intercept) and (slope) are updated to minimize the mean squared error (MSE).

The parameter updates for gradient descent are defined as follows:

where:

- is the learning rate,

- is the gradient of the loss function with respect to .

The gradient of the MSE loss function with respect to parameters  and for a dataset with samples is given by:

Given the above, answer the following:

You are given a simple dataset with the following points:

Assume the initial parameters are  and . Using a learning rate of  , perform one step of gradient descent to update  and .

- A)  = 0.4 , = 1.2
- B)  = 0.6 , = 1.8
- C) = 0.2 ,  = 0.6
- D) = 0.8 ,  = 2.0

15.

If the learning rate  is set too high, what is the most likely consequence for the training process?

- A) The gradient descent will converge faster.
- B) The model will reach the global minimum more accurately.
- C) The gradient descent will overshoot the minimum, causing divergence.
- D) The model's parameters will remain constant.

Solution **C**

16.

You are training a machine learning model using gradient descent with momentum. The current gradient of the loss function with respect to the parameter  at iteration   is        . The learning rate is 0.1, and the momentum coefficient  is 0.9.

The velocity term  from the previous iteration is 0.2. Using the momentum update rule:

What is the updated value of  after this iteration if the initial

-A)0.875

-B) 0.915

-C) 0.945

- D)0.985

Answer:D

17.

You are training a neural network using gradient descent with momentum. You have two scenarios:

Scenario A uses a momentum coefficient β=0.5.

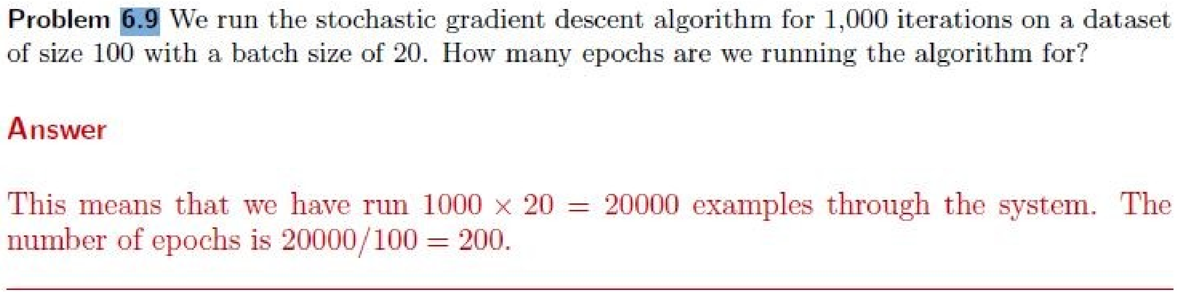
Scenario B uses a momentum coefficient β=0.9.

Assuming all other hyperparameters (learning rate, batch size, initialization, etc.) are the same, which of the following statements is most likely to be true?

A) Scenario A will converge faster but may overshoot the minimum.
B) Scenario B will have more stable convergence but will converge slower.
C) Scenario A will result in a smaller number of iterations but will be noisier.
D) Scenario B is more likely to get stuck in local minima due to higher momentum.

Answer: B

18.

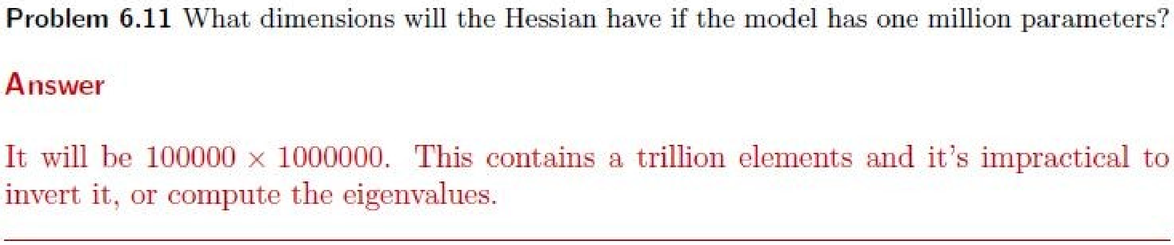


**Problem** 6.9 We run the stochastic gradient descent algorithm for 1,000 iterations on a dataset of size 100 with a batch size of 20. How many epochs are we running the algorithm for?

**Answer**

This means that we have run $1000 \times 20 = 20000$ examples through the system. The number of epochs is $20000/100 = 200$.

19.



**Problem 6.11** What dimensions will the Hessian have if the model has one million parameters?

**Answer**

It will be $100000 \times 1000000$. This contains a trillion elements and it's impractical to invert it, or compute the eigenvalues.

20.

You have a neural network with one hidden layer using the sigmoid activation function  and a squared error loss function. The network's output is  and the target is  Given the loss function:

What is the derivative of the loss with respect to the output of the hidden layer  where

A)

B)

C)

D)

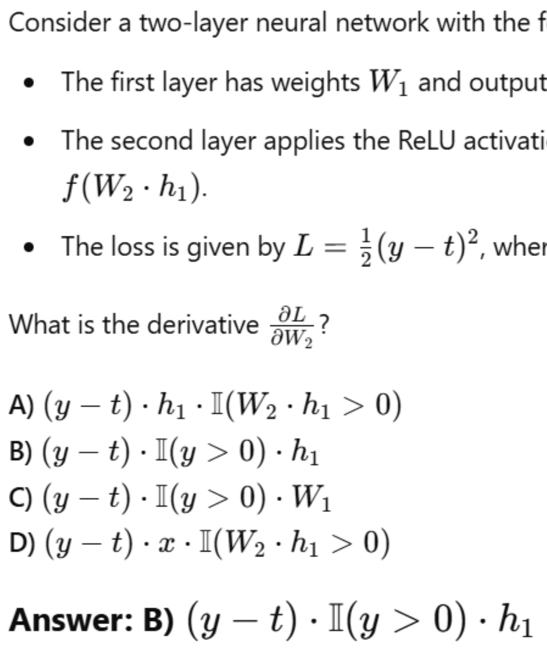
Answer:A)

C is also marked correct

1. Derivative of the loss function with respect to output .

2. The derivative of  with respect to    (due to the sigmoid activation) is .

21.



Consider a two-layer neural network with the following configuration:

- The first layer has weights $W_1$ and output $h_1 = W_1 \cdot x$ (no activation function).
- The second layer applies the ReLU activation function $f(h) = \max(0, h)$ and has output $y = f(W_2 \cdot h_1)$.
- The loss is given by $L = \frac{1}{2}(y - t)^2$, where $t$ is the target.

What is the derivative $\frac{\partial L}{\partial W_2}$?

A) $(y - t) \cdot h_1 \cdot \mathbb{I}(W_2 \cdot h_1 > 0)$
B) $(y - t) \cdot \mathbb{I}(y > 0) \cdot h_1$
C) $(y - t) \cdot \mathbb{I}(y > 0) \cdot W_1$
D) $(y - t) \cdot x \cdot \mathbb{I}(W_2 \cdot h_1 > 0)$

**Answer: B)** $(y - t) \cdot \mathbb{I}(y > 0) \cdot h_1$

22.

Why is He Initialization particularly useful for deep neural networks that use ReLU (Rectified Linear Unit) or its variants as activation functions?

A) It prevents the vanishing gradient problem by setting all weights to zero.
B) It helps maintain the variance of activations in each layer by scaling weights based on the number of input neurons.
C) It reduces overfitting by adding random noise to the initial weights.
D) It ensures that the gradients in backpropagation will always be positive, avoiding the exploding gradient problem.

**Answer: B)**

23.

If a neural network layer has input neurons and you are applying He Initialization, how are the weights typically initialized?

A)

B)

C)

D)

Answer: A

24.

What happens to the model's performance when the model is underfitting?

A) It performs well on training data but poorly on test data
B) It performs poorly on both training and test data
C) It performs well on test data but poorly on training data
D) It performs well on both training and test data

**Answer: B**

**25.**

What is the trade-off between bias and variance in a deep learning model?
B) As bias decreases, variance increases
C) As bias increases, variance increases
D) As bias decreases, variance decreases

Answer: B

26.

Which of the following statements is true regarding variance?

A) High variance models are likely to underfit the training data.

B) Variance measures the sensitivity of the model to small fluctuations in the training dataset.

C) Lower variance is always preferred, regardless of bias.

D) Variance and bias are unrelated concepts.

Answer: B)

27.

What is the effect of high variance in a machine learning model?

A) The model performs consistently on different datasets.

B) The model learns noise in the training data, leading to overfitting.

C) The model is too simplistic and fails to capture underlying patterns.

D) The model provides biased estimates of the target variable.

Answer: B)

28.

Given an input image of size 32×32 with 3 channels, a convolutional layer with 8 filters of size 5×5 stride = 1, and padding = 0, what will be the output size?

A) 32×32×8

B) 28×28×8

C) 30×30×8

D) 26×26×8

Answer: B) 28×28×8

29

An input image has a size of 64×64 with 3 channels. You use a convolutional layer with 16 filters of size 3×3, stride = 1, and padding = 1. What will be the output dimensions?

A) 64×64×16

B) 62×62×16

C) 60×60×16

D) 66×66×16

Answer: A) 64×64×16

30.

An input of size 128×128×3 goes through a convolutional layer with 16 filters of size 3×3, stride = 1, and padding = 1, followed by another convolutional layer with 32 filters of size 3×3, stride = 1, and padding = 1. What is the output size after both layers?

A) 128×128×32

B) 126×126×32

C) 64×64×32

D) 128×128×16

Answer: A) 128×128×32

31.If a convolutional layer has an input of size 64×64×3, uses 64 filters each of size 3×3, stride = 1, and padding = 1, how many trainable parameters does the layer have?

A) 1,792

B) 1,792,000

C) 1,728

D) 1,728,000

Answer: A) 1,792
(Calculation: (3×3×3+1)×64=1,792

32.

What is the impact of increasing the stride in a 1D or 2D convolution?

A) Increases the resolution of the output

B) Reduces the depth of the output feature maps

C) Reduces the spatial dimensions of the output

D) Increases the number of learnable parameters

Answer: C)

33.

Why might you choose "same" padding instead of "valid" padding in a convolutional layer?

A) To decrease the computational complexity

B) To maintain the same output size as the input size

C) To increase the number of filters in the layer

D) To reduce the number of parameters in the layer

Answer: B)

34.

What is a primary use of 1x1 convolutions in 2D ConvNets?

A) To downsample the input spatially

B) To increase the spatial resolution

C) To reduce the number of channels while keeping the spatial dimensions intact

D) To perform non-linear transformations

Answer: C)

Q:

Consider a neural network with the following structure:

- **Input layer**: 4 neurons (representing the 4 input features)
- **First hidden layer**: 6 neurons

- **Second hidden layer**: 4 neurons
- **Output layer**: 2 neurons

Calculate the total number of parameters (including biases) for this network.

Answer: 68

Q:

Consider a  neural network with the following structure:

Input layer: 2 neurons (representing the 2 input features)

First hidden layer: 3 neurons

Second hidden layer: 5 neurons

Output layer: 1 neuron

calculate the number of parameters for the above network, (bias included )

Answer: 35