In a binary classification problem use a Bayesian approach to update your belief about the parameter 0, which represents the probability of classifying an instance as positive (class 1). The prior distribution for 0 is a Beta distribution with parameters a-3 and B-5.

You observe the following data:

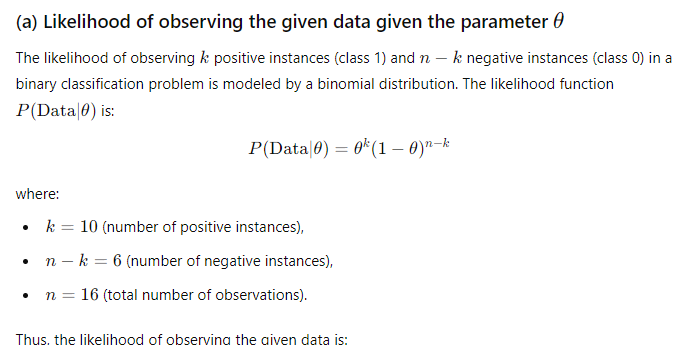
* 10 positive instances (class 1)
* 6 negative instances (class 0

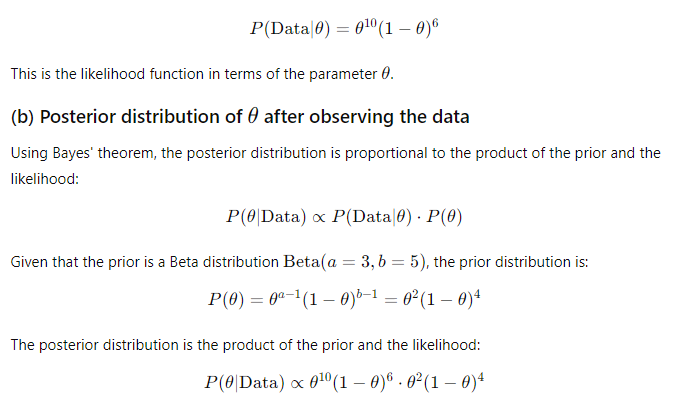
Calculate the following:

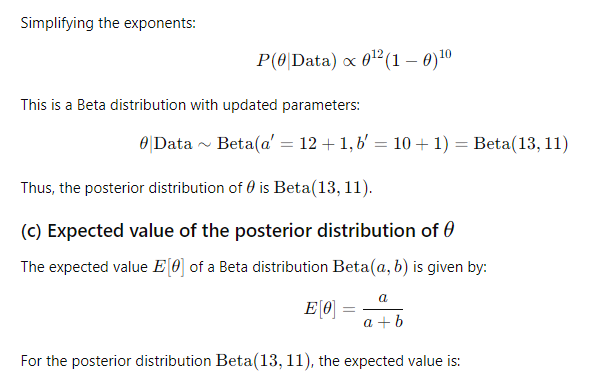
a) Determine the likelihood of observing the given data given the parameter 0.

b) Find the posterior distribution of 0 after observing the data.

c) Calculate the expected value of the posterior distribution of 0.









A deep neural network is designed for a binary classification task with the following architecture:

* Input layer: 120 features
* Hidden Layer 1: 150 neurons, using ReLU activation
* Hidden Layer 2: 80 neurons, using ReLU activation
* Output Layer: 1 neuron, using Sigmoid activation

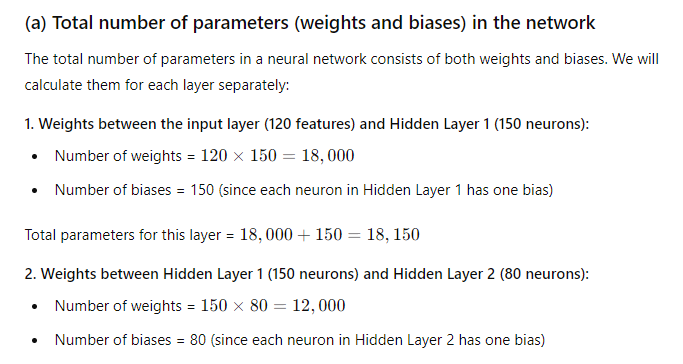
The network is trained using stochastic gradient descent (SGD) with a batch size of 64 and a learning rate of 0.005. The loss function used is binary cross-entropy.

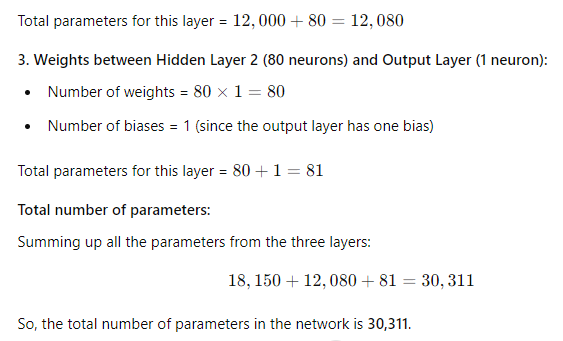
Calculate the following:

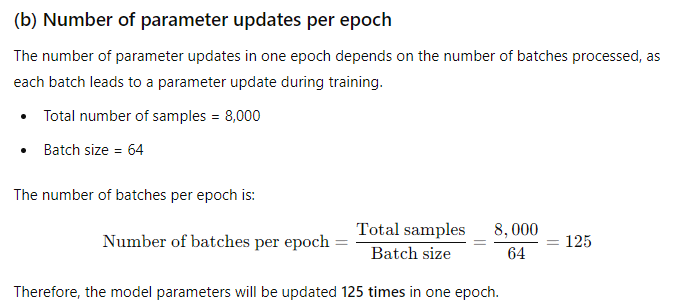
a) Determine the total number of parameters (weights and biases) in this network.

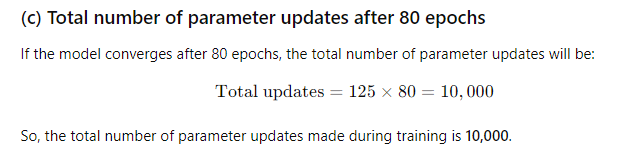
b) If the training dataset contains 8,000 samples, calculate how many times the model parameters will be updated in one epoch.

c) If the training process converges after 80 epochs, calculate the total number of parameter updates made during training.









A recurrent neural network (RNN) for sentiment analysis was designed using a dataset of 6,000 movie reviews, each with a maximum length of 150 words. The reviews are categorized into positive or negative sentiment. The dataset is split into training and validation sets with a 85:15 ratio. The RNN architecture includes:

* Input layer
* Embedding layer with an embedding size of 150
* LSTM layer with 50 units
* Dense layer with 1 unit using a sigmoid activation function for binary classification.

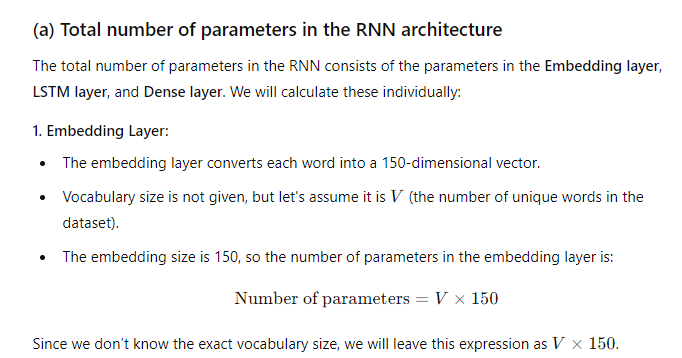
Assuming you use binary cross-entropy as the loss function and the Adam optimizer, compute the following:

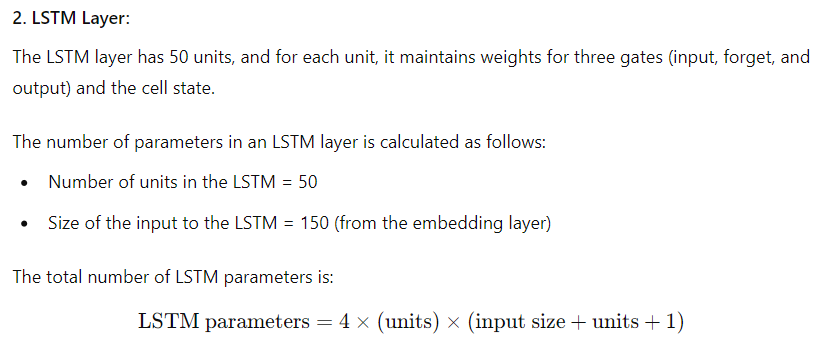
a) Calculate the total number of parameters in this RNN architecture.

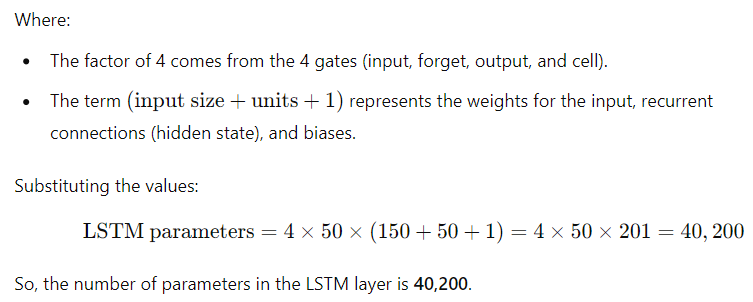
b) Determine the shape of the output from the LSTM layer.

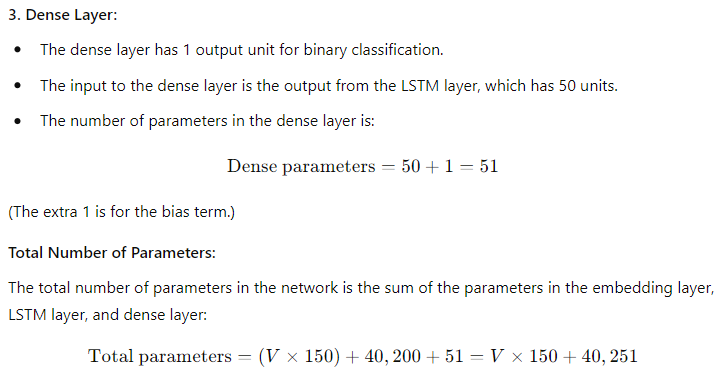
c) Find the batch size if you want to process the entire training set in one epoch using mini-batch gradient descent.

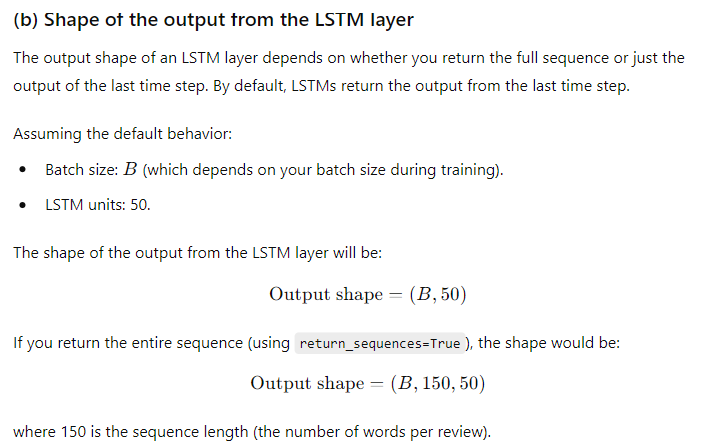
d) If the validation loss starts to increase during training, suggest at least two methods to address overfitting and enhance the model's generalization.

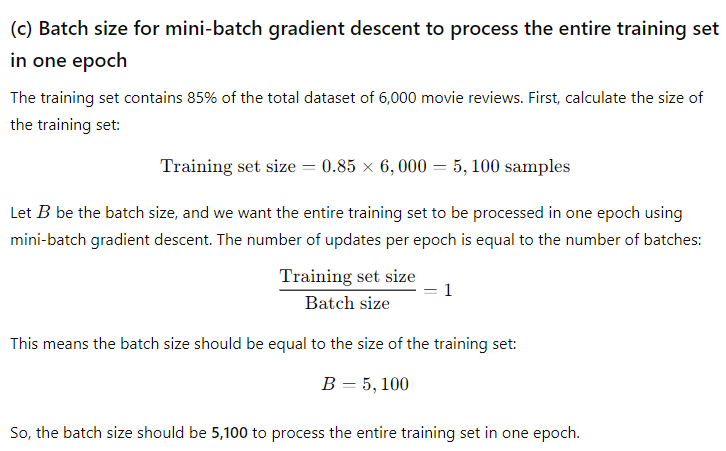


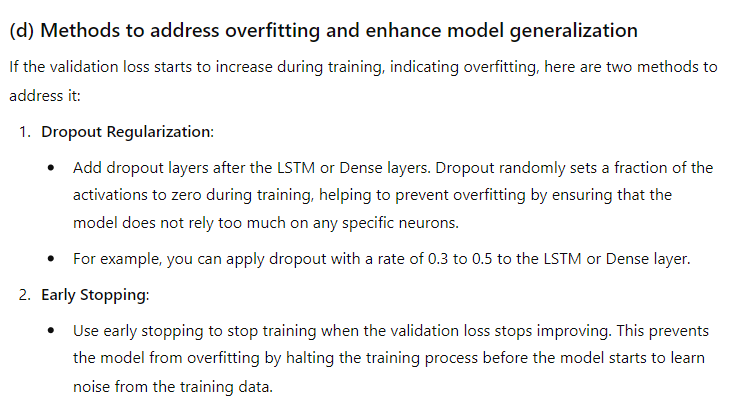








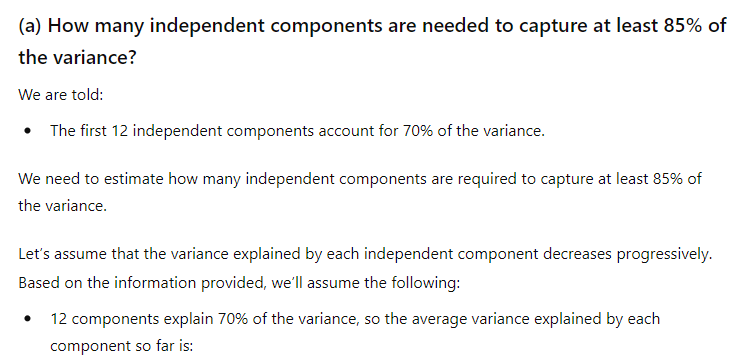


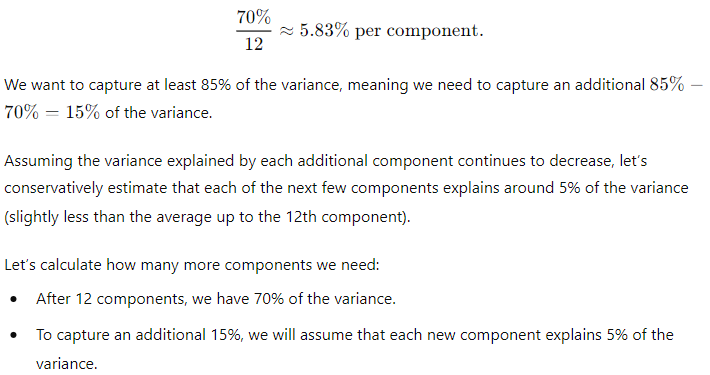


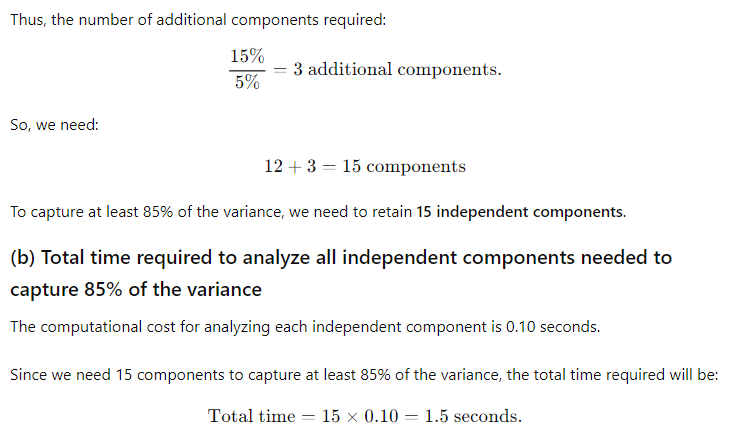
.A dataset comprising 1,800 samples of chemical compounds, where each sample is described by a vector of 300 features. You apply Independent Component Analysis (ICA) to identify underlying independent components from this dataset.

a) After performing ICA, you determine that the first 12 independent components account for 70% of the total variance in the dataset. To capture at least 85% of the variance, how many independent components will you need to retain?

b) If the computational cost for analyzing each independent component is 0.10 seconds, calculate the total time required to analyze all the independent components needed to capture 85% of the variance.







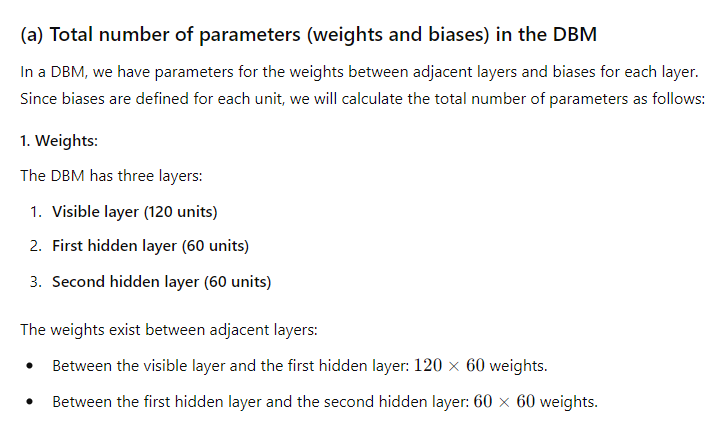
Consider a Deep Boltzmann Machine (DBM) with the following configuration:

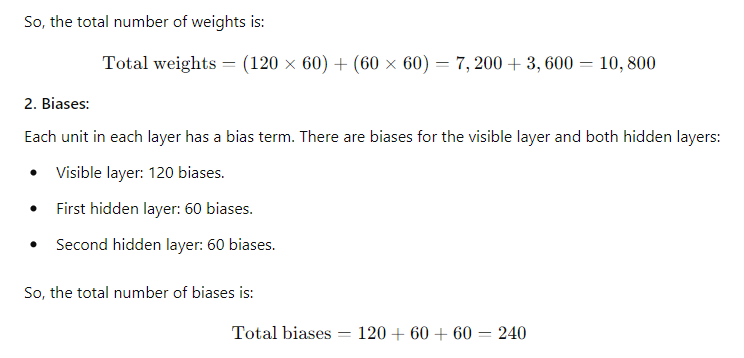
* Network Architecture:
* Visible Layer: 120 units
* Two Hidden Layers: Each with 60 units
* The DBM is trained using Contrastive Divergence (CD) with a mini-batch size of 64 and a learning rate of 0.005.

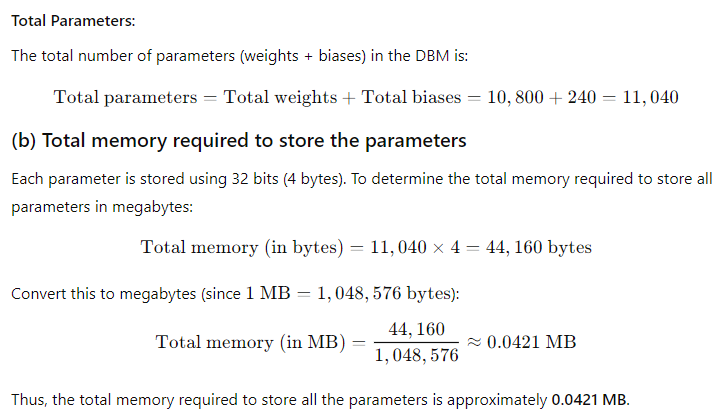
a) Calculate the total number of parameters (weights and biases) in the DBM

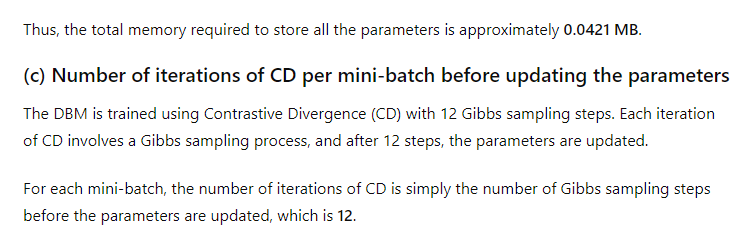
b) Each parameter is stored using 32 bits. Determine the total memory required (in megabytes) to store all the parameters of the DBM.

c) If the CD algorithm is run for 12 Gibbs sampling steps, calculate how many iterations of CD are performed for each mini-batch before updating the parameters.









6.Analyze all thefeat Analyze the performance of a deep learning model based on its training and validation loss curves over multiple epochs:

Epoch 1: Training Loss = 2.2, Validation Loss = 1.9

Epoch 2: Training Loss = 1.9, Validation Loss = 1.8

Epoch 3: Training Loss = 1.8. Validation Loss = 2.0

Epoch 4: Training Loss = 1.6, Validation Loss = 2.

Epoch 5: Training Loss 1.4, Validation Loss = 2.8

Epoch 6: Training Loss 1.3, Validation Loss = 3.

Epoch 7: Training Loss = 1.2, Validation Loss = 3.6

Epoch 8: Training Loss 1.1, Validation Loss = 4.0

a) Determine the epoch at which the model begins to overfit based on the loss curves.

b) Describe a strategy to mitigate overfitting in this scenario

c) If the training loss continued to decrease while the validation loss increased, explain the potential impact on the model's performance on unseen data.

REFER CLASS NOTES:

A convolutional neural network (CNN) was trained for image classification on a dataset of 12,000 images. To optimize your model, you use a validation set for hyperparameter tuning. You split your dataset into 75% training data and 25% validation data,Your CNN has the following hyperparameters:

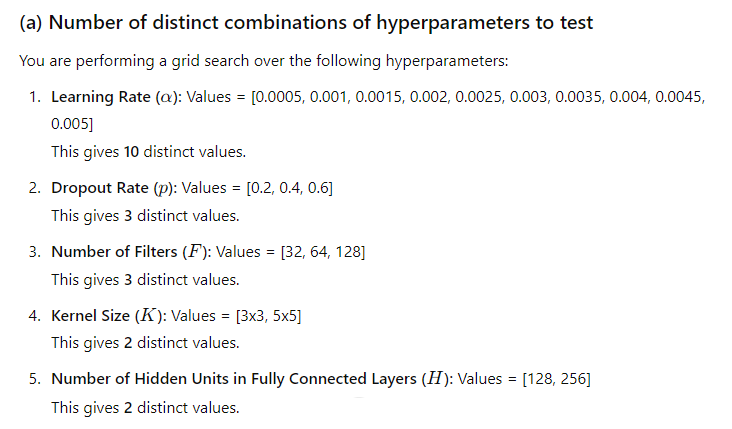
* Learning Rate (a): 0.0005 to 0.005 in increments of 0.0005
* Dropout Rate (p): 0.2 to 0.6 in increments of 0.2
* Number of Filters (F): 32, 64, or 128
* Kernel Size (K): 3x3 or 5x5
* Number of Hidden Units in Fully Connected Layers (H): 128 or 256

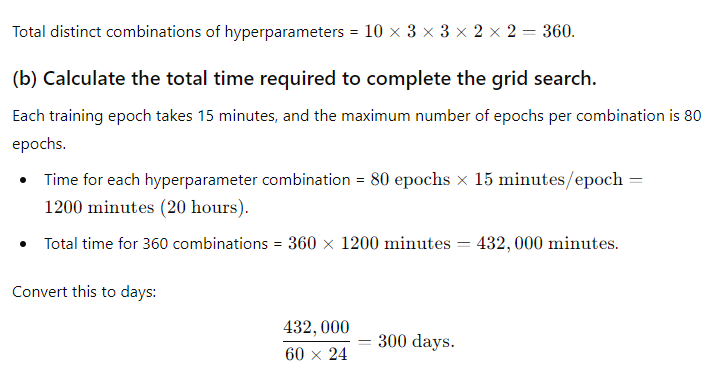
You decide to use a grid search to find the best hyperparameter combination based validation accuracy. on

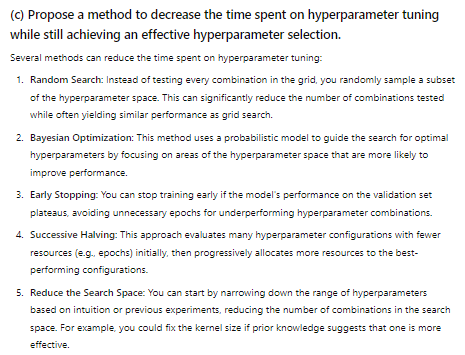
a) Determine how many distinct combinations of hyperparameters you will test.

b) If each training epoch takes 15 minutes and you limit the grid search to a maximum of 80 epochs per hyperparameter combination, calculate the total time required to complete the grid search.

c) Propose a method to decrease the time spent on hyperparameter tuning while still achieving an effective hyperparameter selection.

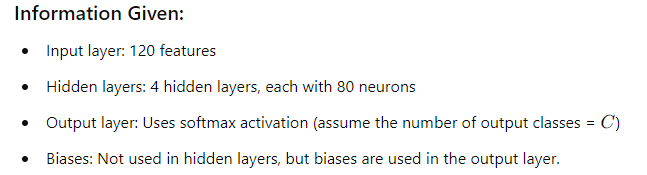


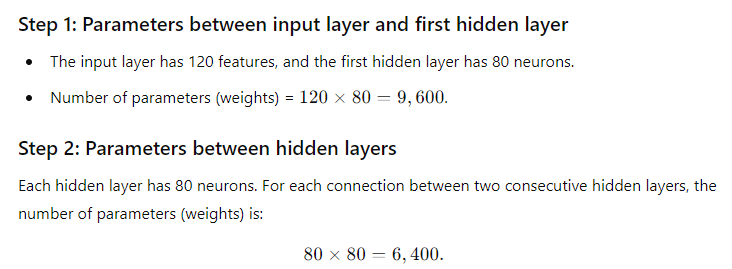


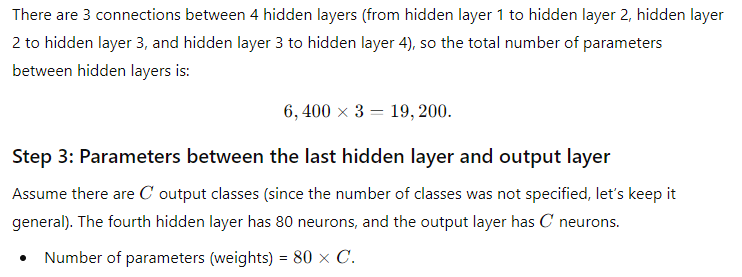


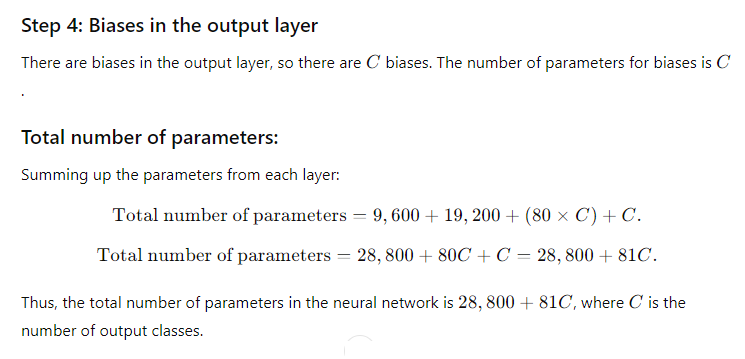
Consider a deep neural network with 4 hidden layers, where each layer has 80 neurons. The network utilizes the ReLU activation function in the hidden layers anda softmax activation function in the output layer for classification. The loss functionemployed is categorical cross-entropy.

Given a training dataset with 8,000 samples, where each sample has 120 features, calculate the total number of parameters in the neural network. Assume that biases are not used in the hidden layers, and include the weights and biases in the output layer in your calculation.









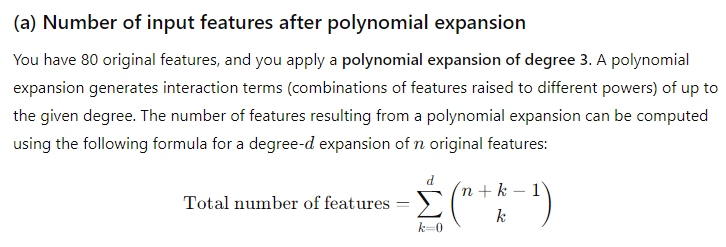
Consider a dataset consisting of time series measurements with 80 features,recorded over 600 time steps. You plan to use Slow Feature Analysis (SFA) to identify the slowest features from this dataset. You decide to extract the top 6 slowest features.

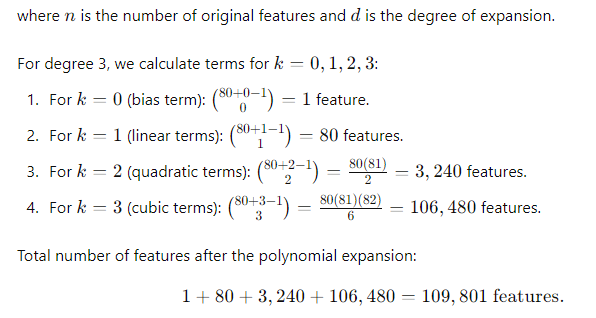
a) If you apply a polynomial expansion of degree 3 to create the input space for SFA, how many input features will result from this expansion?

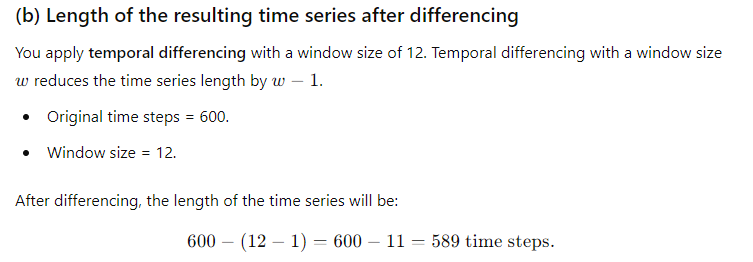
b) Given a window size of 12 for temporal differencing in SFA, what will be the length of the resulting time series after differencing?

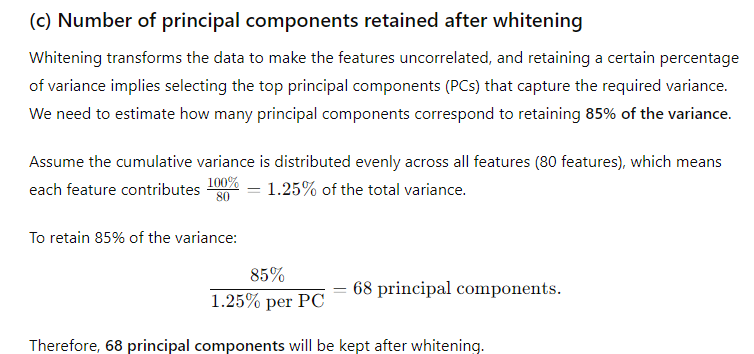
c) Before performing SFA, you apply a whitening step to the data and retain 85% of the variance. How many principal components will be kept after this step?

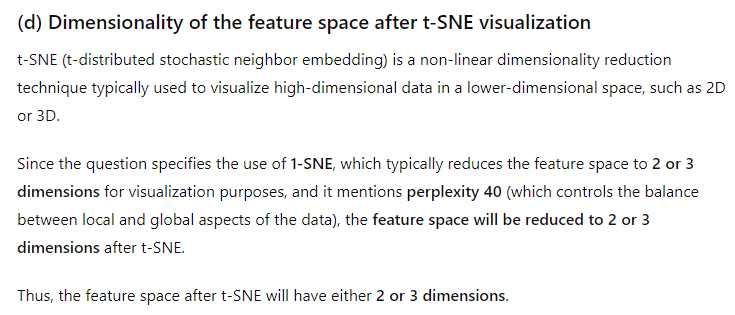
d) After executing SFA, if you want to visualize the learned features using 1-SNE with a perplexity of 40, how many dimensions will the feature space have?









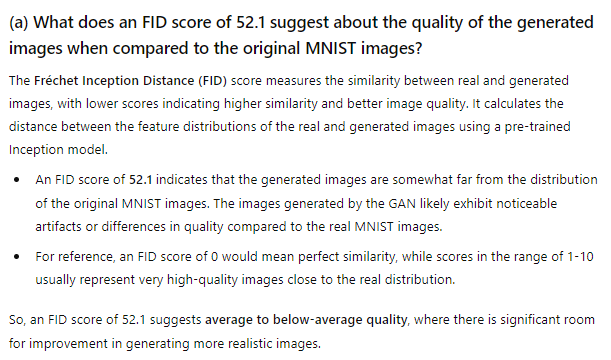


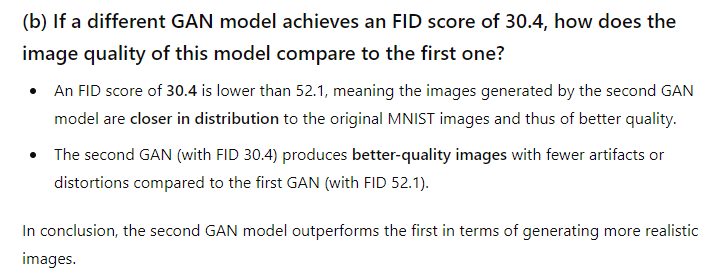
A Generative Adversarial Network (GAN) is trained to generate images of handwritten digits similar to those in the MNIST dataset. The quality of the generated images is assessed using the Fréchet Inception Distance (FID) score.After training, the FID score for the GAN is recorded as 52.1.

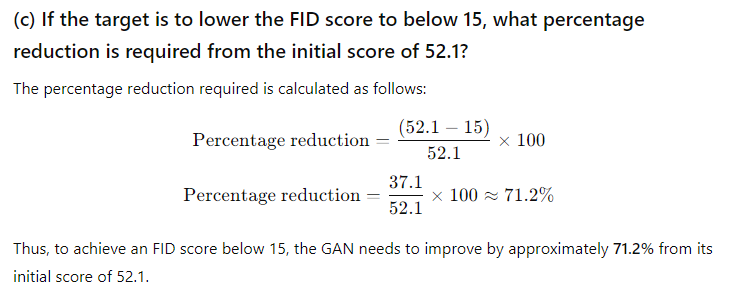
a) What does an FID score of 52.1 suggest about the quality of the generated images when compared to the original MNIST images?

b) If a different GAN model achieves an FID score of 30.4, how does the image quality of this model compare to the first one?

c) If the target is to lower the FID score to below 15, what percentage reduction is required from the initial score of 52.1?







Consider a deep neural network with the following architecture for a binary classification task:-- Input layer with 100 features Two hidden layers with 200 and 100 neurons respectively, both using ReLU activation Output layer with a single neuron using Sigmoid activation The network is trained using stochastic gradient descent (SGD) with a batch size of 32 and a learning rate of 0.01. The loss function used is binary cross-entropy.

a) Calculate the total number of parameters (weights and biases) in this network.

b) If the training dataset contains 10,000 samples, how many times will the model parameters be updated in one epoch?

c) If the training process converges after 100 epochs, how many parameter updates would have been made in total?

**(a) Calculate the total number of parameters (weights and biases) in this network.**

The network architecture is as follows:

* Input layer: 100 features
* Hidden layer 1: 200 neurons (ReLU activation)
* Hidden layer 2: 100 neurons (ReLU activation)
* Output layer: 1 neuron (Sigmoid activation)

**Step 1: Weights and biases between input layer and hidden layer 1**

* Number of weights: 100×200=20,000
* Number of biases: 200 (since each neuron in the hidden layer has one bias term)

**Total parameters for input to hidden layer 1:**

20,000 weights+200 biases=20,200 parameters

**Step 2: Weights and biases between hidden layer 1 and hidden layer 2**

* Number of weights: 200×100=20,000
* Number of biases: 100

**Total parameters for hidden layer 1 to hidden layer 2:**

20,000 weights+100 biases=20,100 parameters

**Step 3: Weights and biases between hidden layer 2 and output layer**

* Number of weights: 100×1=100
* Number of biases: 1

**Total parameters for hidden layer 2 to output layer:**

100 weights+1 bias=101

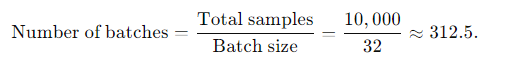
**Total number of parameters**

Summing all the parameters from the three steps:

20,200+20,100+101=40,401 total parameters.

**(b) How many times will the model parameters be updated in one epoch?**

* The training dataset contains **10,000 samples**.
* The batch size is **32**



**(c) How many parameter updates would have been made in total after 100 epochs?**

The total number of parameter updates is simply the number of updates per epoch multiplied by the number of epochs:

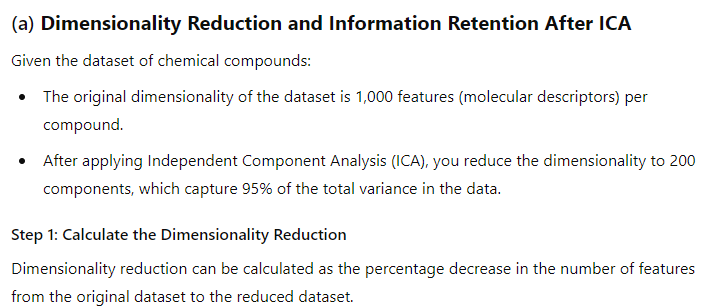
Total updates=312 updates per epoch×100 epochs=31,200 total updates.

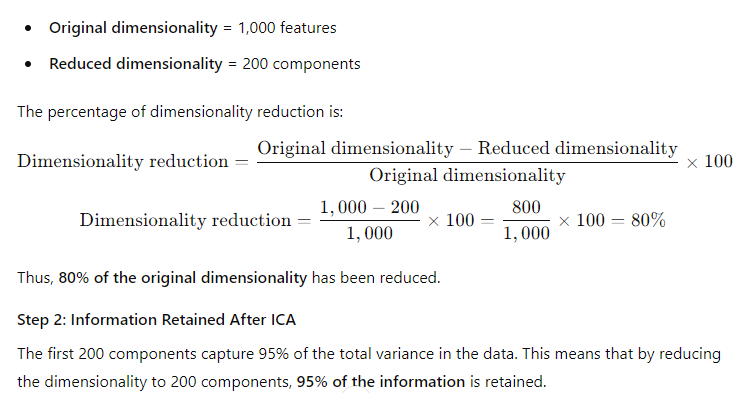
Thus, **31,200 parameter updates** would have been made in total after 100 epochs.

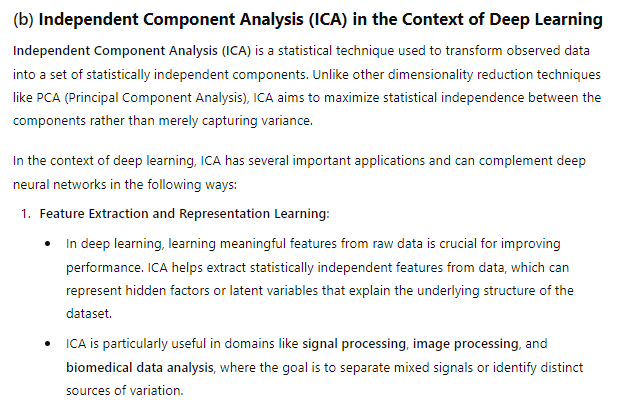
Consider a dataset of chemical compounds for drug discovery, where each compound is represented by a set of molecular descriptors. The dataset contains 5,000 compounds, and each compound is described by 1,000 independent features. You are tasked with applying Independent Component Analysis (ICA) to extract meaningful independent features from the compound dataset. After applying ICA, the resulting independent components show that the first 200 components capture 95% of the total variance in the data.

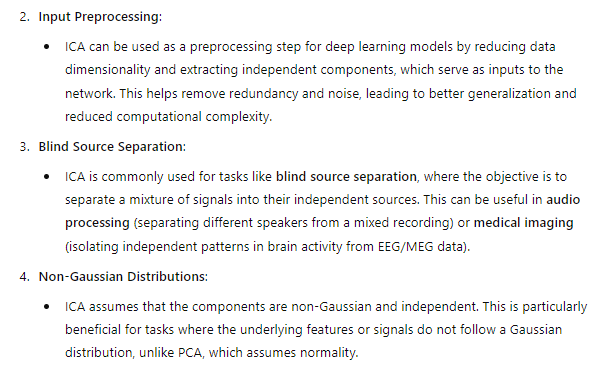
a) Calculate the dimensionality reduction achieved by ICA and determine the percentage of information retained after reducing the dimensionality to the first 200 components.

b) Describe the concept of Independent Compound Analysis in the context of deep learning.







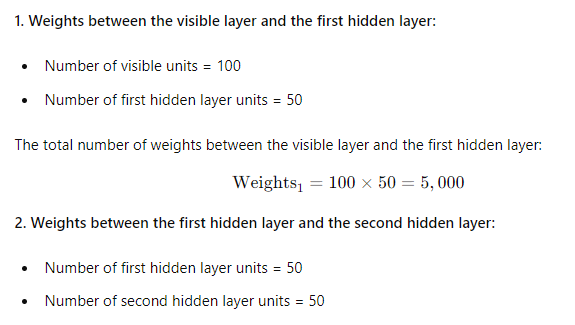


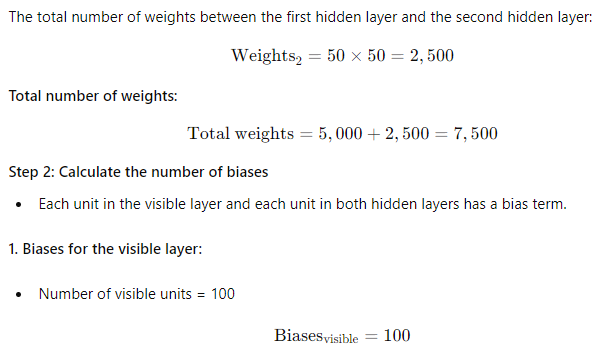
Consider a Deep Boltzmann Machine (DBM) with two layers of hidden units and one visible layer. The visible layer has 100 units, and each hidden layer has 50 units. The DBM is trained using Contrastive Divergence (CD) with a mini-batch size of 32 and a learning rate of 0.01.

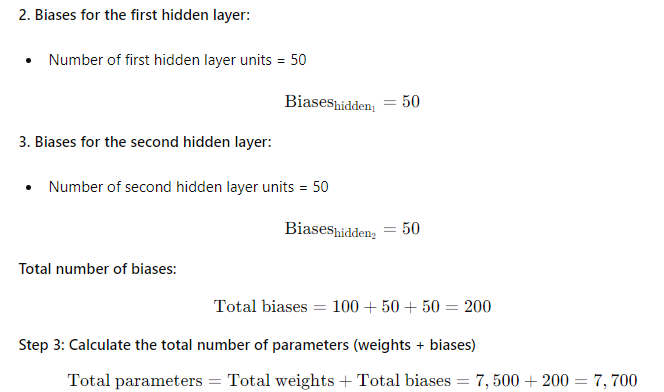
a) How many parameters (weights and biases) are there in the DBM?

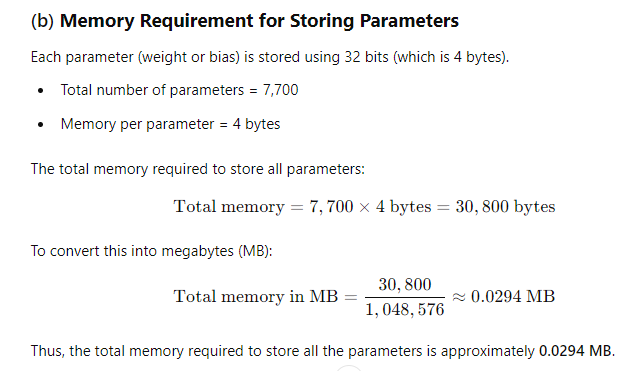
b) If each parameter is stored using 32 bits, calculate the total memory (in megabytes) required to store all the parameters of the DBM.

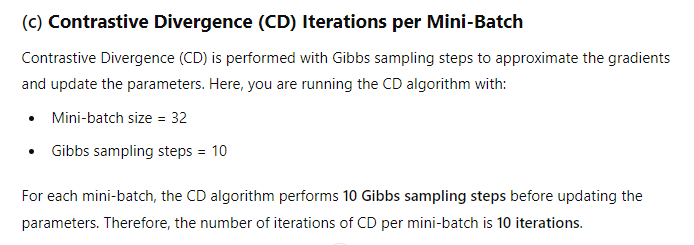
c) During training, if the CD algorithm is run for 10 Gibbs sampling steps, how many iterations of CD are performed for each mini-batch before updating the parameters?











a) A machine learning model is being trained on a dataset with 1000 samples. The dataset is divided into training, validation, and test sets with a ratio of 70:15:15. The model has two hyperparameters: learning rate and regularization strength.

1. The learning rate is chosen from the set {0.001, 0.01, 0.1}.

2. The regularization strength is chosen from the set {0.01, 0.1, 1.0}. If we use grid search to find the best combination of hyperparameters using the validation set, how many different combinations of hyperparameters do we need to try?

b) A machine learning model is being trained on a dataset with 2000 samples. The dataset is divided into training, validation, and test sets with a ratio of 60:20:20. The model has three hyperparameters: learning rate, number of hidden units, and batch size.

1. The learning rate is chosen from the set {0.001, 0.01, 0.1}.

2. The number of hidden units is chosen from the set {50, 100, 200}.

3. The batch size is chosen from the set {32, 64, 128}. If we use grid search to find the best combination of hyperparameters using the validation set, how many different combinations of hyperparameters do we need to try?

**TRY YOURSELF**

Consider training a deep neural network with 10 hidden layers, each containing 100 neurons. The network is trained using mini-batch gradient descent with a batch size of 64. The activation function used in each layer is the ReLU function. The dataset consists of 50,000 samples.

a) Calculate the total number of parameters in the network, including weights and biases.

b) If each parameter is stored using 32 bits, calculate the total memory (in megabytes) required to store all the parameters of the network.

c) If the network takes 100 epochs to converge, calculate the total number of weight updates performed during training. Assume that each weight and bias parameter is stored separately

**TRY YOURSELF**