TASK-1: Neural network models using Keras for the ‘Regression task’

I. Description of Dataset: eye_movements 43946

- **Task:** The main task of the dataset is classification. Specifically, it involves categorizing individual words within sentences into one of three classes: irrelevant, relevant but not answering the question, or the correct answer to the question. This task is derived from a larger context of eye movement research, where eye tracking data is used to understand cognitive processes, such as reading comprehension and information retrieval. Given this dataset, the classification task involves using the provided features (eye movement metrics) to predict the relevance of each word to the question, as represented by the classification labels. This task is essential for understanding how eye movements relate to cognitive processes such as comprehension and information retrieval during reading tasks.
- **Features:** The dataset contains 22 features extracted from eye movement data. These features capture various aspects of eye movements while participants read sentences. Some of the key features include:
  - ***Fixations:*** Information about the number of fixations on a word, duration of fixations, and whether the fixation occurred during the first encounter of the word.
  - ***Saccades:*** Lengths of the first and last saccades, which are rapid eye movements between fixations.
  - ***Positions:*** Distances between fixations and word positions within the sentence.
  - ***Pupil Diameter:*** Measurements of maximum pupil diameter during fixations.
  - ***Regression:*** Information about regressions, including the number of regressions initiated from the word and their durations.
  - ***Other Metrics:*** Additional metrics such as total fixation duration, mean fixation. duration, and pupil diameter lag.
- **Target:** Each word in the dataset is labeled with a classification label indicating its relevance to the question presented in the assignment. The classification labels are as follows:
  - ***0 (Irrelevant):*** Indicates that the word is irrelevant to the question.
  - ***1 (Relevant):*** Indicates that the word is relevant to the question but does not answer it directly.
  - ***2 (Correct Answer):*** Indicates that the word is the correct answer to the question.

II. Clear description of the four models

Each model has a distinct architecture in terms of the number and arrangement of layers, as well as the activation functions used, to tackle the regression problem on the dataset provided:

i. Model 1: Simple Two-Layer Network

- **Architecture:** This model consists of two dense layers. The first layer has 64 nodes, and the second layer has 32 nodes.
- **Activation Function:** Both layers use the ReLU (Rectified Linear Unit) activation function. ReLU is chosen for its ability to avoid the vanishing gradient problem and accelerate the convergence during training.
- **Output Layer:** The output layer consists of a single neuron with a linear activation function to predict a continuous output value, suitable for regression tasks.

ii. Model 2: Three-Layer Network

- **Architecture:** This model is slightly more complex, with three dense layers. The configuration starts with 100 nodes in the first layer, followed by 50 in the second, and 25 in the third layer.
- **Activation Function:** Like Model 1, this model employs the ReLU activation function for all hidden layers. This choice supports efficient learning and generalization across more complex network structures.
- **Output Layer:** It also ends with a single neuron with a linear activation function for predicting the output.

iii. Model 3: Equal Three-Layer Network

- **Architecture:** It features three dense layers, each having 30 nodes. This equal distribution is aimed at maintaining a uniform complexity throughout the network.
- **Activation Function:** The sigmoid activation function is used for all layers. Unlike ReLU, sigmoid can map the input values into a 0-1 range, which might be beneficial depending on the nature and scale of the output values expected.
- **Output Layer:** Concludes with a single neuron and a linear activation function for output prediction.
-

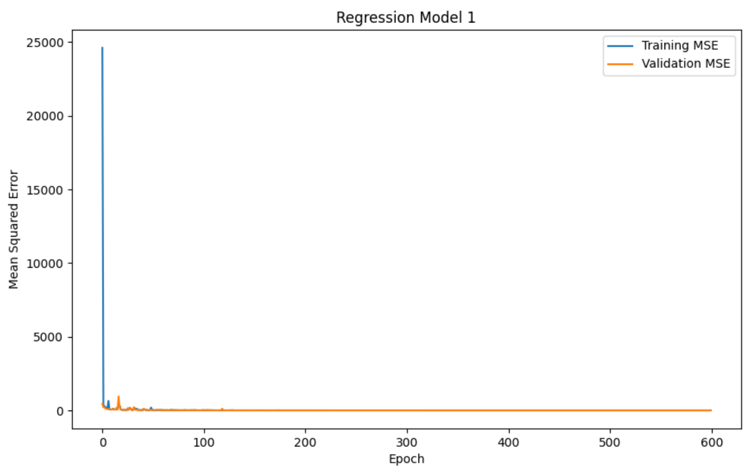
iv. Model 4: Four-Layer Network

- **Architecture:** This is the most complex model among the four, with four layers having descending units: 50, 25, 10, and 5 respectively.

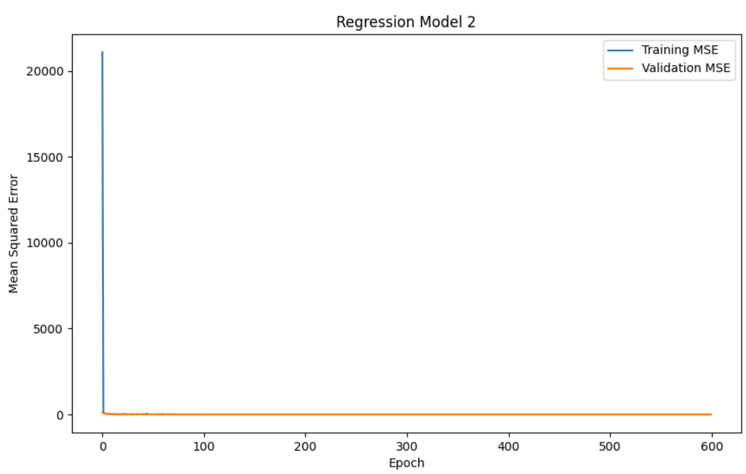
- **Activation Function:** This model uses the tanh (hyperbolic tangent) activation function for all hidden layers. Tanh functions like a scaled sigmoid activation but outputs values ranging from -1 to 1, which can provide smooth gradients and potentially better handling of negative inputs.
- **Output Layer:** Features a single output neuron with a linear activation function.

III. Four graphs, one for each model

i. Model 1: Simple Two-Layer Network graph

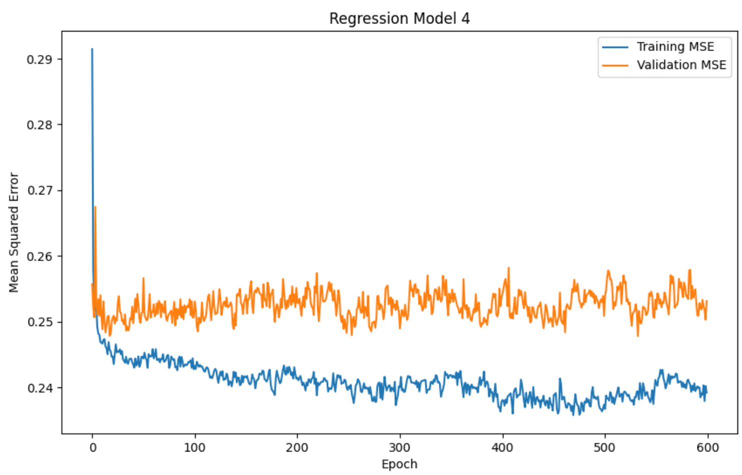


ii. Model 2: Three-Layer Network graph



iii. Model 3: Equal Three-Layer Network

iv. Model 4: Four-Layer Network



IV. A table of minimum validation errors

Sr. No.	Model	Minimum Validation Error
1	Model 1	0.247053
2	Model 2	0.249552
3	Model 3	0.248778
4	Model 4	0.247803

V. Discussion of the results

Interpreting the Plots:

1. ***Regression Model 1 and 2:*** The plots display a very steep decline in the MSE for the training set initially, which quickly stabilizes. This behavior is typical for models that are learning quickly from the data. The validation MSE also drops significantly and flattens out, which suggests that the model is generalizing well beyond the training data. The similarity in the shapes of the training and validation curves suggests that there's no significant overfitting.

2. ***Regression Model 3 and 4:*** These models show a different pattern with a much finer scale on the y-axis, suggesting that the errors are smaller overall. There is more variance in the validation MSE, which could indicate that the models are more sensitive to the data they are validating against. The fluctuations in the validation MSE also suggest that the model could be learning more nuanced aspects of the data but may also be more prone to variance in the validation set.

Analyzing the Table:

- ***Model 1:*** Achieves a minimum validation error of 0.247053.
- ***Model 2:*** Has a slightly higher minimum validation error of 0.249552.
- ***Model 3:*** Comes close to Model 1 with a minimum validation error of 0.248778.
- ***Model 4:*** Has a minimum validation error of 0.247803, very close to that of Model 1.

Based on the minimum validation error, ***Model-1*** appears to be the best performing one. However, the differences between the models are quite slight. The choice of the best model may also be influenced by other factors, such as the complexity of the model (which isn't evident from the plots), the training time, the interpretability of the model, and how it may perform on completely unseen test data.

TASK-2: Neural network models using Keras for the 'Classification task'

I. Description of Dataset: USPS 41964

- **Task:** This dataset is also derived from the USPS handwritten digit database and shares the objective of recognizing handwritten digits from images. The task focuses on developing a model to classify these images into their respective digit categories accurately.
- **Features:** Each instance in this dataset consists of a grayscale image of a handwritten digit standardized to a size of 16x16 pixels. Like the previous dataset, the features represent the pixel values of these images. Each feature reflects the intensity of a pixel, resulting in 256 features per sample.
- **Target:** The target variable in this dataset serves the same purpose as in the previous dataset. It indicates the true numeric value of the handwritten digit depicted in each image, with values ranging from '0' to '9', representing the digits in the images. Both datasets involve the same fundamental task of digit recognition from grayscale images of handwritten digits. The features represent pixel intensities, and the target variable denotes the actual numeric values of the handwritten digits in the images. The goal is to train machine learning models to effectively classify these images based on their pixel values and accurately predict the corresponding digits.

II. Clear description of the four models

For the classification task, the models designed using the Keras framework varied in architecture to explore different depths and complexities suitable for the task. Each model has unique specifications in terms of the number of layers and nodes, as well as the activation functions used, to effectively handle binary classification:

i. Model 1: Simple Two-Layer Network

- **Architecture:** This model comprises two dense layers. The first layer has 64 nodes, followed by a second layer with 32 nodes.
- **Activation Function:** Both hidden layers use the ReLU (Rectified Linear Unit) activation function. ReLU is commonly used in classification tasks for its efficiency and effectiveness in facilitating gradient propagation in deeper models.
- **Output Layer:** The output layer has a single neuron with a sigmoid activation function. This setup is ideal for binary classification, as the sigmoid function maps the output between 0 and 1, which can be interpreted as the probability of the input belonging to the positive class.

ii. **Model 2: Three-Layer Network**

- **Architecture:** More complex than Model 1, this model includes three layers with descending numbers of nodes: 100, 50, and 25.
- **Activation Function:** The ReLU activation function is utilized for all hidden layers, maintaining efficient learning and preventing vanishing gradients.
- **Output Layer:** A single output neuron with a sigmoid activation is used to predict the binary outcome, making it suitable for tasks where a binary decision is required.

iii. **Model 3: Uniform Three-Layer Network**

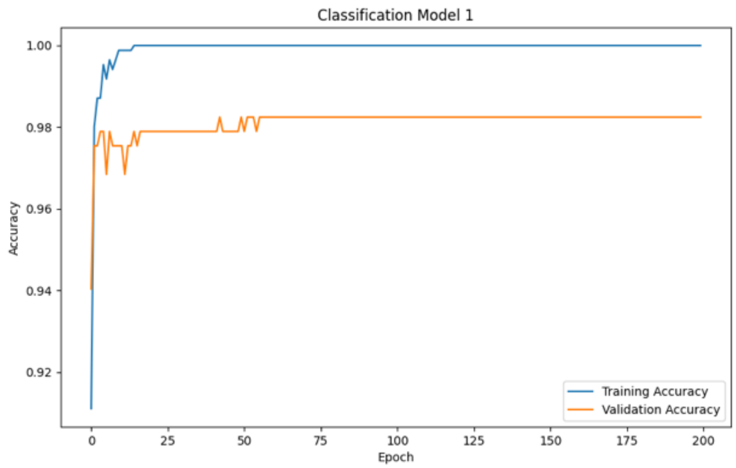
- **Architecture:** This model employs three layers, each containing 30 nodes, which creates a balanced complexity across the network.
- **Activation Function:** The sigmoid activation function is used for all layers. Unlike ReLU, sigmoid compresses the outputs between 0 and 1 at every layer, which can affect how features are represented internally but might benefit certain datasets by emphasizing non-linear separability.
- **Output Layer:** It concludes with a single neuron and a sigmoid activation, consistent with binary classification needs.

iv. **Model 4: Four-Layer Network**

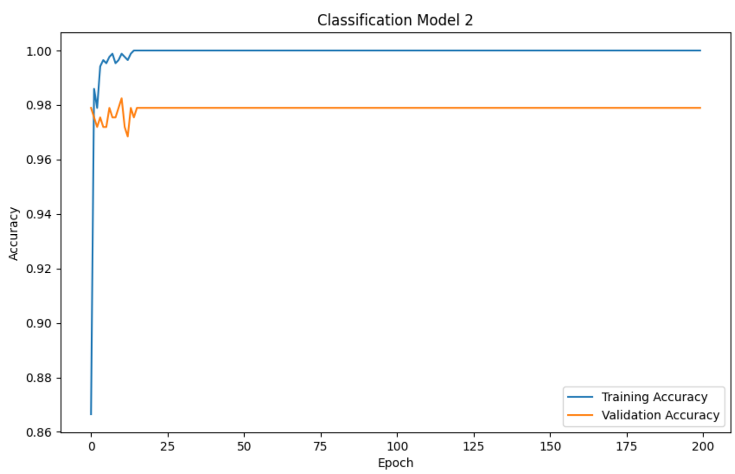
- **Architecture:** The most complex model in this set, featuring four layers with 50, 25, 10, and 5 nodes respectively, arranged to gradually reduce the dimensionality and refine the features extracted.
- **Activation Function:** All hidden layers use the tanh (hyperbolic tangent) activation function. Tanh outputs values from -1 to 1 and can provide a different scale of feature normalization which might be beneficial in some classification scenarios.
- **Output Layer:** Includes a single output neuron with a sigmoid activation, suitable for outputting the probability of class membership.

III. Four graphs, one for each model

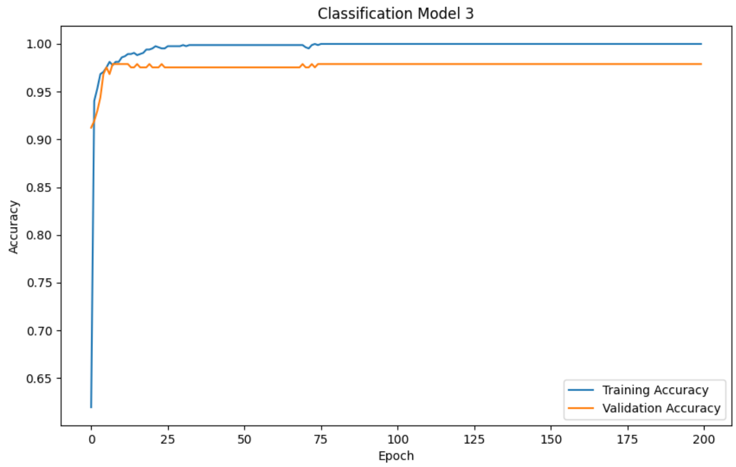
i. Model 1: Simple Two-Layer Network



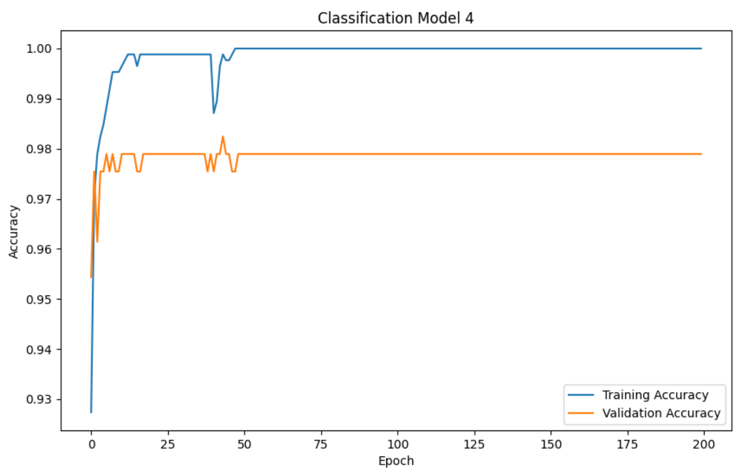
ii. Model 2: Three-Layer Network



iii. Model 3: Uniform Three-Layer Network



iv. Model 4: Four-Layer Network



IV. A table of minimum validation errors

Sr. No.	Model	Maximum Validation Accuracy
1	Model 1	0.982456
2	Model 2	0.982456
3	Model 3	0.978947
4	Model 4	0.982456

V. Discussion of the results

Interpreting the Plots:

1. ***Classification Model 1:*** The training accuracy quickly reaches a high level and then remains relatively stable. The validation accuracy also stabilizes at a high level but with slight fluctuations. This suggests that the model generalizes well and does not overfit significantly to the training data.

2. ***Classification Model 2:*** Like Model 1, the training accuracy rapidly ascends to a plateau. The validation accuracy is also high but experiences more variability, which might be a sign of the model trying to adapt to the validation set's patterns without losing generalization.

3. ***Classification Model 3:*** This model demonstrates an initial sharp increase in training accuracy, followed by stability with minimal fluctuations. However, the validation accuracy shows more pronounced fluctuations, which might indicate that the model is less stable in terms of generalization compared to Models 1 and 2.

4. ***Classification Model 4:*** The training accuracy for this model quickly reaches a high value and maintains it with minimal variance. The validation accuracy also achieves high levels but with some variability, suggesting a good but slightly less consistent generalization performance compared to Model 1.

Analyzing the Table:

- Model 1 has a maximum validation accuracy of 0.982456.
- Model 2 has the same maximum validation accuracy of 0.982456.
- Model 3 has a slightly lower maximum validation accuracy of 0.978947.
- Model 4 also reaches a maximum validation accuracy of 0.982456.

Models 1, 2, and 4 all share the highest maximum validation accuracy, while Model 3 is marginally lower. This small difference could be within the margin of error for the validation set or may represent a real but minor difference in performance. Given that all models perform within a very tight range of each other, the choice of the "best" model may depend on other factors such as computational efficiency, the complexity of the model, training time, or even the specific nature of the errors it makes.

If overfitting is a concern, you might prefer a model with less fluctuation in validation accuracy, potentially indicating that it generalizes better to unseen data. In that case, Model 1 would seem preferable. If training time or computational resources are constraints, you might choose a model that converges faster to high accuracy. In the absence of other constraints, any of the three models with the highest maximum validation accuracy could be considered equally suitable based on the provided data.