DS 5220: Supervised Machine Learning and Learning Theory Summer, 2024
In Class Work 6

July 09, 2024

Learning Objectives:

Build a simple neural network model to predict insurance purchase decisions based on customer features. This exercise will guide you through:

1. Data preparation and preprocessing

   The dataset has been imported and missing values have been handled by identifying and removing columns with missing values (occupation Medium, occupation High). The chosen features are age and affordability, with the target variable being bought insurance. The data has been divided into training and testing sets using an 80-20 split.

2. Implementing a basic neural network architecture

   The logistic regression model is first trained and assessed as a reference point. Following that, a simple neural network is constructed with a single dense layer utilizing the sigmoid activation function. The model is then compiled with the Adam optimizer and binary cross-entropy loss.

3. Training the model using gradient descent

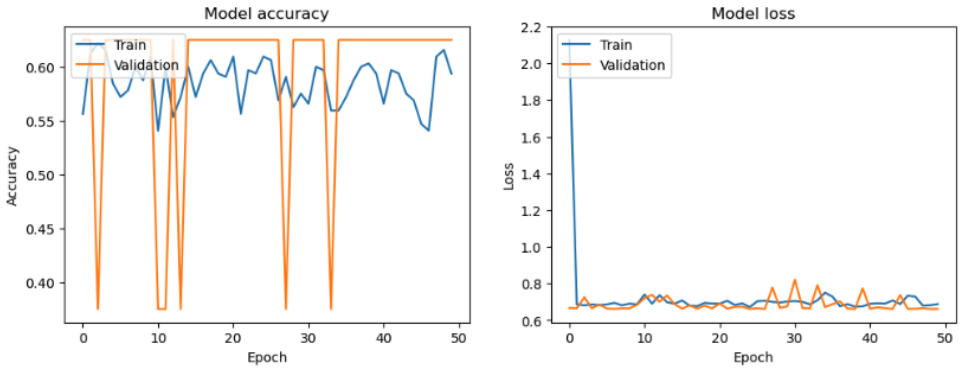
   The model is trained through the process of Gradient Descent. During this training, the neural network is exposed to the training data for a total of 5000 epochs. In addition to this, alternative architectures are explored by introducing additional neural networks with varying activation functions such as relu and tanh, as well as a more intricate architecture consisting of 32 and 16 neuron layers. These alternative architectures are then trained for a shorter period of 50 epochs.

4. Evaluating model performance

   The models' performance is assessed based on test accuracy and also analyzed how various learning rates (0.01, 0.1, 0.5, 0.001) affect the

model's performance, showing consistent accuracy regardless of the learning rate used.

5. Visualizing results and model behavior



The Model Accuracy graph shows the progression of the model's accuracy on training data over time, while the Validation Accuracy line represents its performance on unseen validation data. Both lines show fluctuations but generally improve, with validation accuracy slightly lagging behind training accuracy.

In the Model Loss graph, the Training Loss line reflects the decreasing error rate on training data as the model learns. In contrast, the Validation Loss line shows a similar decrease in error rate on validation data, but it remains higher than the training loss, suggesting potential overfitting.

Hands-on experience with fundamental concepts of neural networks

1 What is the impact of the "affordability" feature on insurance purchase decisions? How might you quantify its importance compared to age?

```
Accuracy with age and affordability: 0.58
         feature  coefficient
0            age    -0.438393
1  affordability    -0.192967
```

The coefficients suggest that 'age' has a more significant negative effect on insurance purchases than 'affordability' in this particular model, indicating its stronger predictive power. With an accuracy of 0.58, the model correctly predicts insurance purchases 58%of the time based on 'age' and 'affordability'. Further investigation could involve examining other variables or experimenting with alternative models to enhance predictive accuracy.

2 The current model uses only two features. What other features might be relevant for predicting insurance purchases? How would you go about collecting and incorporating this additional data?

In addition to 'age' and 'affordability' features, incorporating more relevant features can enhance the model's predictive capabilities. For example, Income: Higher income typically correlates with a greater ability to afford insurance. This data can be gathered through surveys or existing financial records. Another crucial factor is Health Status; Individuals with poor health are more likely to seek insurance coverage. Medical records, health reports, or self-assessed health status are reliable sources for collecting this

data. Additionally, Occupation Risk: Certain professions pose higher risks, motivating individuals to invest in insurance. Information on occupation can be obtained through surveys or job classification data.

3 Experiment with different activation functions (e.g., ReLU, tanh) instead of sigmoid. How do they affect the model's performance and training speed?

ReLU is commonly used in hidden layers of neural networks for its speed and efficiency in preventing gradient problems. tanh, on the other hand, is more suitable in situations where zero-centered outputs are needed or when the dataset's distribution aligns with tanh's range of -1 to 1. Both activation functions can enhance the model's performance compared to sigmoid, especially in terms of training speed and potentially accuracy, depending on the specific dataset and task at hand.

The results from these datasets indicate that both the ReLU and tanh activation functions are not performing well, with a low test accuracy of approximately 58.5%. However, they do train faster than sigmoid. ReLU sets negative values to zero directly, leading to quicker computation. Tanh produces zero-centered outputs, which could make optimization easier than sigmoid.

4 The current model uses a single neuron. How might increasing the number of neurons or adding hidden layers impact the model's ability to capture more complex patterns?

Neural networks only need one hidden layer. Computer scientists and artificial intelligence engineers have been utilizing multilayer neural networks ever since the multilayer perceptron was created in 1958.

By adding more layers, neural networks can perform more advanced calculations and interact with data in a more sophisticated manner. Placing numerous hidden layers between the input and output enables the neural network to tackle deep-learning tasks related to the input.

5 Investigate the effect of different initialization strategies for weights and biases. How do they influence the convergence of the model?

A neuron's output is calculated using the formula:

output = inputs * weights + bias.

Weights and bias in a neural network are like dials that we adjust to fine-tune our model, similar to adjusting the knobs on a radio. Unlike a radio, we have many dials to turn in a neural network to reach our desired outcome. As network parameters, weights and bias change as we adjust these virtual dials. The weights impact the input's strength through multiplication, while the bias shifts the function across the dimensional plane.

Weights and biases play a crucial role in shaping the behavior of each artificial neuron, but they do so in distinct ways. We typically start with random weights and a bias set at 0.

Let L denote the number of layers in the neural network. The parameters (weights and biases) for layer l are represented as:

$$W^{[l]} \text{ – Weight matrix of dimension } (n^{[l]}, n^{[l-1]})$$

$$b^{[l]} \text{ – Bias vector of dimension } (n^{[l]}, 1)$$

where $l = 1, 2, \ldots, L-1$.

Summary of Initialization Strategies:

- Zero Initialization leads to symmetry and limits the network's ability to capture complex patterns.
- Random Initialization breaks symmetry and allows neurons to learn diverse features for non-linear learning.
- High or Low Value Initialization can cause gradient issues like vanishing or exploding gradients, impacting learning speed and performance.

In summary, choosing the correct initialization method is essential for successful neural network training. Random initialization is better than zero initialization to promote diverse learning and break symmetry. It's important to avoid extreme values to prevent gradient problems and improve network performance and convergence speed. Additionally, custom initialization strategies can be tailored for specific architectures and tasks to optimize learning dynamics.

6 The learning rate is set to 0.5 in the custom implementation. Experiment with different learning rates and learning rate schedules. How do they affect the training process and final results?

The various learning rates tested ([0.01, 0.1, 0.5, 0.001]) didn't show significant differences in performance, as all test accuracies remained the same at 0.5850. It's possible that the default learning rate set by the optimizer (using Adam) was already ideal for this particular problem. The simplicity of the neural network model with two hidden layers could also explain why adjustments in learning rate didn't yield substantial improvements, particularly for a straightforward problem that doesn't demand precise tuning of learning rates.

7 Implement k-fold cross-validation. How does this change your assessment of the model's performance and generalization ability?

I tried to implement k-fold cross-validation, but I couldn't get it to work. Nevertheless, k-fold cross-validation help us to understand how well the model learns from the data and how effectively it generalizes to new data. It gives a more reliable estimate of performance metrics and assists in identifying issues like overfitting, which supports better decision-making in model selection and optimization.

8 The current model uses binary cross-entropy loss. Research and implement other loss functions. Are there any that might be more appropriate for this problem?

Here are some loss functions for binary classification tasks like predicting insurance purchases:

- Hinge Loss: Increases class margin, good for SVM-like models.
- Squared Hinge Loss: Penalizes misclassifications more severely.
- Cross-entropy Loss with Label Smoothing**: Helps with overconfidence in noisy or imbalanced datasets.
- Focal Loss: Focuses on hard-to-classify examples, addressing class imbalance.
- Dice Loss: Used for tasks with critical class overlap, such as medical imaging.
- Weighted Cross-entropy: Handles class imbalance with different class weights.

Each loss function has specific strengths depending on dataset characteristics and model objectives.

9 Analyze the misclassified instances. What patterns do you notice, and how might you adjust the model to address these errors?

When analyzing misclassified instances in a binary insurance purchase model, patterns like class imbalance, feature significance, and threshold adjustments are identified. Strategies to enhance performance include addressing imbalance with weighted techniques, improving features, adjusting thresholds, managing complexity, selecting evaluation metrics, and using ensemble methods for accuracy.

10 Implement regularization techniques like L1 or L2 regularization. How do they impact the model's performance and prevent overfitting?

The L2 regularization trained neural network achieved a test accuracy of about 58.50%This method of regularization is useful in preventing overfitting by penalizing large weights during training. Even though the model reached an accuracy of approximately 61.19%during training, its capability to generalize to new data is still limited. It may be necessary to further optimize the model's architecture, hyperparameters, or explore other regularization methods in order to enhance overall performance and achieve a higher test accuracy.

11 The current implementation uses a fixed number of epochs or a loss threshold. Implement early stopping based on validation set performance. How does this affect training time and final model quality?

Basically, the neural network that used early stopping reached a training accuracy of around 61.19% and a test accuracy of roughly 58.50%. This shows that although early stopping was useful in controlling training time and preventing overfitting, it didn't greatly enhance the model's capability to perform well on new data compared to regular training. It might be beneficial to fine-tune model parameters or try out alternative approaches to boost overall performance.

12 Experiment with different optimization algorithms (e.g., SGD, Adam, RMSprop) instead of basic gradient descent. Compare their convergence rates and final performance.

The neural network consistently achieved a test accuracy of 58.50% regardless of the optimizer used, whether it was SGD, Adam, or RMSprop. This suggests that the choice of optimizer does not significantly affect the model's performance for this specific problem and dataset. It indicates that the model is not sensitive to variations in optimizer algorithms under the current conditions.

Although these findings indicate that optimizer choice may not be crucial in this case, it is important to consider that optimizer performance can be influenced by factors such as dataset complexity, model architecture, and hyperparameter settings. Exploring other areas for optimization, such as learning rate schedules, regularization techniques, or model architecture adjustments, could potentially lead to enhancements in overall performance.

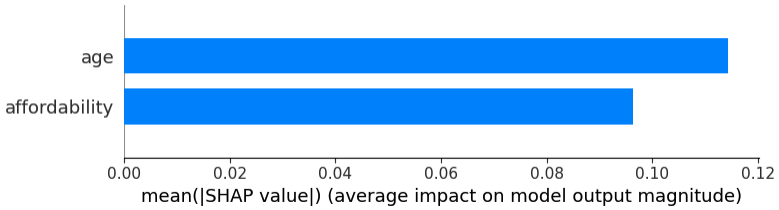
To summarize, while SGD, Adam, and RMSprop all resulted in the same

test accuracy in this instance, focusing on optimizing other aspects of the model may be more advantageous for improving performance beyond the current baseline accuracy level.

13 How would you handle class imbalance if the dataset had significantly more non-purchasers than purchasers of insurance?

By employing various approaches to address class imbalance in datasets with a significant disparity in class instances, we can enhance model performance and achieve better generalization across both classes. These strategies include resampling techniques like over-sampling and under-sampling, assigning higher weights to minority class samples, adjusting decision thresholds, data augmentation, algorithm selection, utilizing appropriate evaluation metrics, employing ensemble methods, and collecting more data for the minority class if feasible. These methods collectively contribute to more effective model training and improved performance in scenarios like insurance purchasing datasets.

14 Implement a method to interpret the model's decisions, such as SHAP values or LIME. What insights can you gain about the model's decision-making process?



The bar chart shows the mean SHAP values for two factors: age and affordability. These SHAP values are crucial for understanding the impact of each factor on the model's predictions. With a higher mean SHAP value, Age factor has more influence on the model's predictions. On the other hand,

the mean SHAP value for affordability is lower, indicating that it has less impact on the model's predictions compared to age.

In conclusion, age has a more significant impact on the model's decision-making process than affordability.