Capstone weekly report#4

First of all, in reference to the question raised in report1, it's essential to clarify the use-case or the selling point of this project. This model serves the purpose of predicting trends in the food and restaurant industry. This prediction capability is valuable for restaurants as it aids in designing new food recipes and formulating advertising strategies.

Upon scrutinizing the previous work, I identified errors in the data splitting and data loading processes. Specifically, the code for splitting data into specific portions did not function correctly with the target dataset, and I neglected to load the data from AWS onto my local machine. I have since rectified these issues, and now the dataset is appropriately split and loaded.

Furthermore, I have been diligently working on refining the model architecture and exploring the best activation function, loss function (considering the project's error costs), and evaluation methods for the food classification project. I'm also determining the optimal hyperparameters, including callbacks such as checkpoints, early stopping, and learning rates. This ongoing process necessitates extensive research and study, which I will continue in the coming week.

In the model architecture, I incorporated the pretrained Resnet50 model on the training dataset with the activation function set to ReLU. Additionally, I introduced custom layers, including batch normalization, dense layers, and possibly a global average pooling layer. I referenced "Convolutional Neural Networks Explained" to calculate the input shape for each layer. However, I encountered an error indicating that the input shape for the global average pooling layer is incorrect.

During my research and study, several questions have arisen:

Q: why using nonlinear activation function for food classification?

A: Nonlinear activation functions enable neural networks to break away from the limitations of linear models and capture complex, nonlinear relationships in data. A general problem with both the sigmoid and tanh functions (linear activation functions) is that they saturate.

Q: What does saturate mean?

A: it means during a range of input x no matter what the value of x is the output of y is a constant. See the graph below when x less than -4 and greater than 4, the value of f(x) is almost 0 or 1. Same thing as tanh activation function, as x becomes very negative or positive, the value of f(x) is close to -1 or 1. When f(x) is a constant, its derivative becomes zero. Consequently, the product of these small gradients (chain rule) results in vanishing gradients, which means that weight updates during training become extremely small. This can lead to slow convergence during training.

A graph with a line drawn on it

Description automatically generated A graph with a line drawn on it

Description automatically generated

Q: Is the value of derivative of f(x) equal to the value of weights?

A: No. the derivative of f(x) is invovled in the euqation of gradient descent. When calculate the gradient by derivate f(x), when the gradient is too small, there is no much to update on weights in training process.

Q: what is gradient descent equation?

A:

A math equation with a triangle and triangle

Description automatically generated

A math equations on a white background

Description automatically generated

Q: what is cross-entropy loss function?

A: the cross-entropy loss function is to calculate the dissimilarity between the predicted probability distribution and the actual probability distribution. It means the less the value of the function the better the model.

A white paper with black text

Description automatically generated

Q: why cross-entropy loss function is good for classification task?

A: when using cross-entropy we usually use one-hot encoded the classes/labels into a vector. After passing a data point through a classification model, the model generates a probability distribution over the classes using a SoftMax activation function. This distribution represents the model's confidence in each class. For example, it might output [0.1, 0.2, 0.6, 0.05, 0.05], indicating that the model is most confident in class 3.

Related reading:

1. <https://machinelearningmastery.com/cross-entropy-for-machine-learning/#:~:text=)%3A%203.288%20bits-,Cross%2DEntropy%20as%20a%20Loss%20Function,be%20used%20for%20classification%20tasks>.
2. Cross entropy and backpropagation: <https://www.youtube.com/watch?v=xBEh66V9gZo>
3. Cross entropy: <https://www.youtube.com/watch?v=6ArSys5qHAU>
4. Customized data augmentation function: <https://github.com/google-research/sam/blob/main/autoaugment/autoaugment.py>
5. Food image classification and image retrieval based on visual features and machine learning. <https://link.springer.com/article/10.1007/s00530-020-00673-6>
6. Deep neural network for food image classification and nutrient identification: A systmenatic review. <https://link.springer.com/article/10.1007/s11154-023-09795-4>’