**Experimental Analysis of different Strategies with Deep Active Learning**

Name: Fady Essam ID: 20190370

Name: Belal Ashraf ID: 20190137

The code represents an implementation of active learning strategies in TensorFlow/Keras for the MNIST and CIFAR-10 datasets with deep neural network model, active learning is a machine learning technique that involves iteratively selecting a subset of unlabeled data points to be labeled by an oracle, such as a human annotator, to improve the performance of the model. The active learning strategies implemented in this code are uncertainty sampling, margin sampling, and entropy sampling.

The code first loads the MNIST and CIFAR-10 datasets using the Keras API. The datasets are split into training and testing sets, and the training data is further split into labeled and unlabeled data. The labeled data is used to train a deep neural network model with convolutional and dense layers.

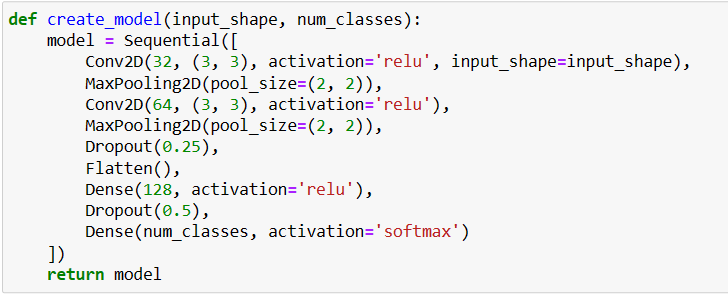
Next, the active learning strategies are implemented as functions that take the model and the unlabeled data as input and return the indices of the top 10 most uncertain, least confident, or most entropic data points, depending on the strategy. The uncertainty sampling strategy selects the data points with the highest predicted class probabilities, while the margin sampling strategy selects the data points with the smallest difference between the highest and second-highest predicted class probabilities. The entropy sampling strategy selects the data points with the highest entropy, which is a measure of the uncertainty of the predicted class probabilities.

The active\_learning function takes the chosen strategy, the training and testing data, the number of initially labeled samples, and the number of iterations as input. In each iteration, the function selects the top 10 data points according to the chosen strategy and adds them to the labeled data. The model is then trained on the new labeled data and evaluated on the test data. The function returns the overall accuracy of the model on the test data after all iterations.

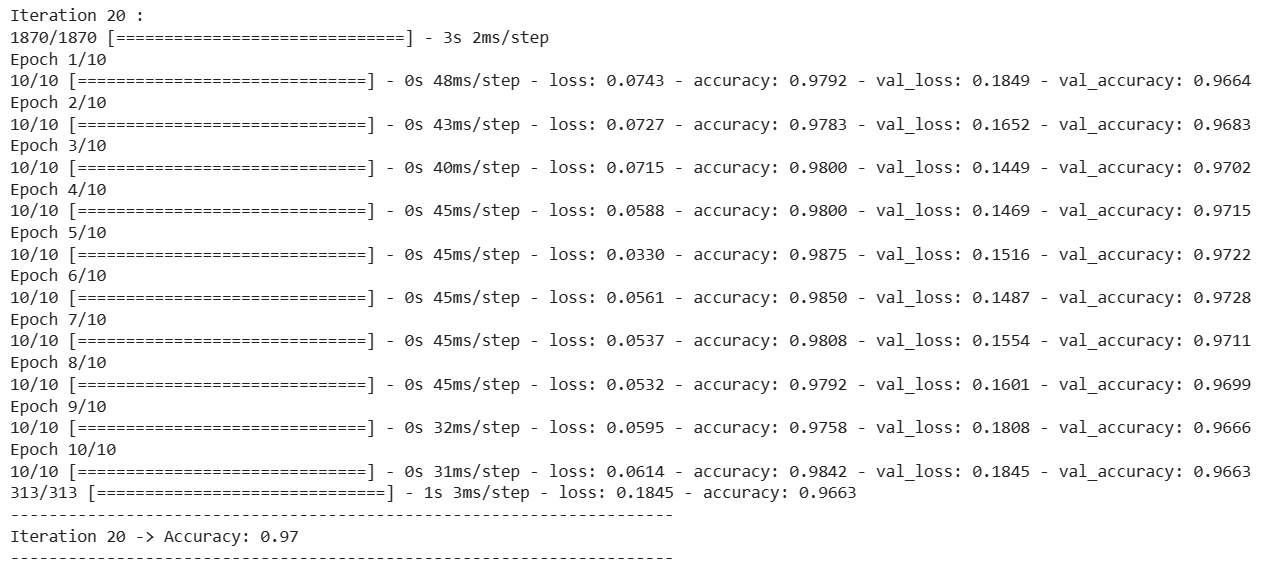
Finally, the active\_learning function is called for each strategy and dataset, and the results are stored in a pandas DataFrame that includes the dataset name, strategy, and accuracy.

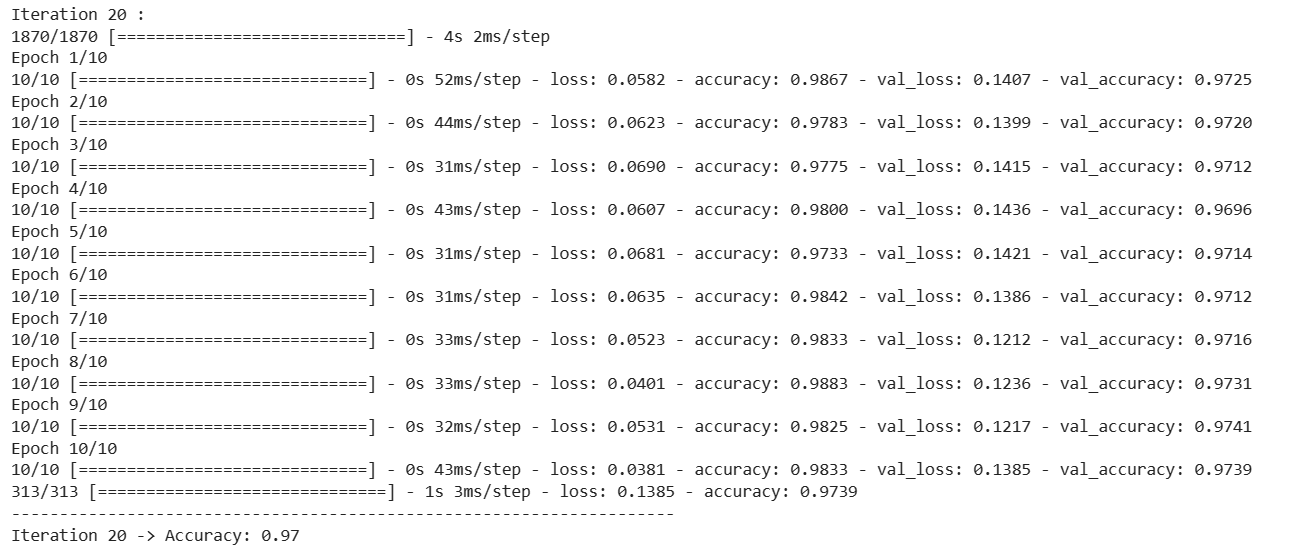
Overall, this code implements deep active learning on two datasets using a various deep neural network models and sampling strategies, and prints the accuracy of each combination after each iteration.

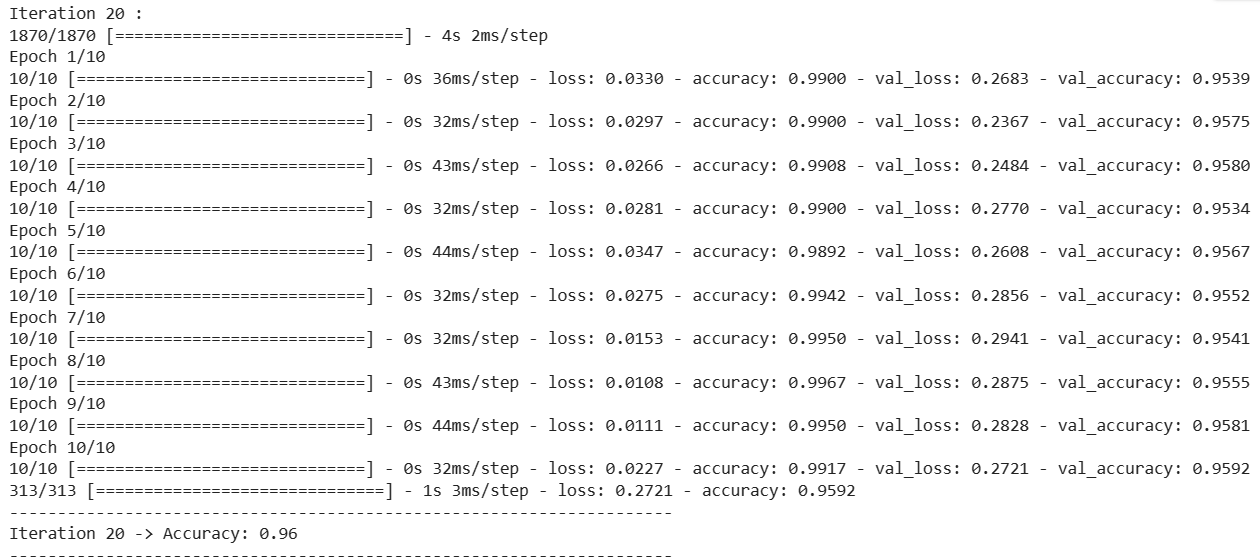
**The Model we use Neural Network model:**



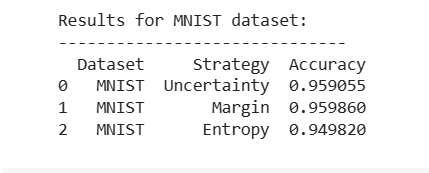
**Accuracy of MNIST dataset after 20 iterations on this model with the 3 strategies:**

* with Uncertainty Strategy
* with Margin Strategy



* with Entropy Strategy

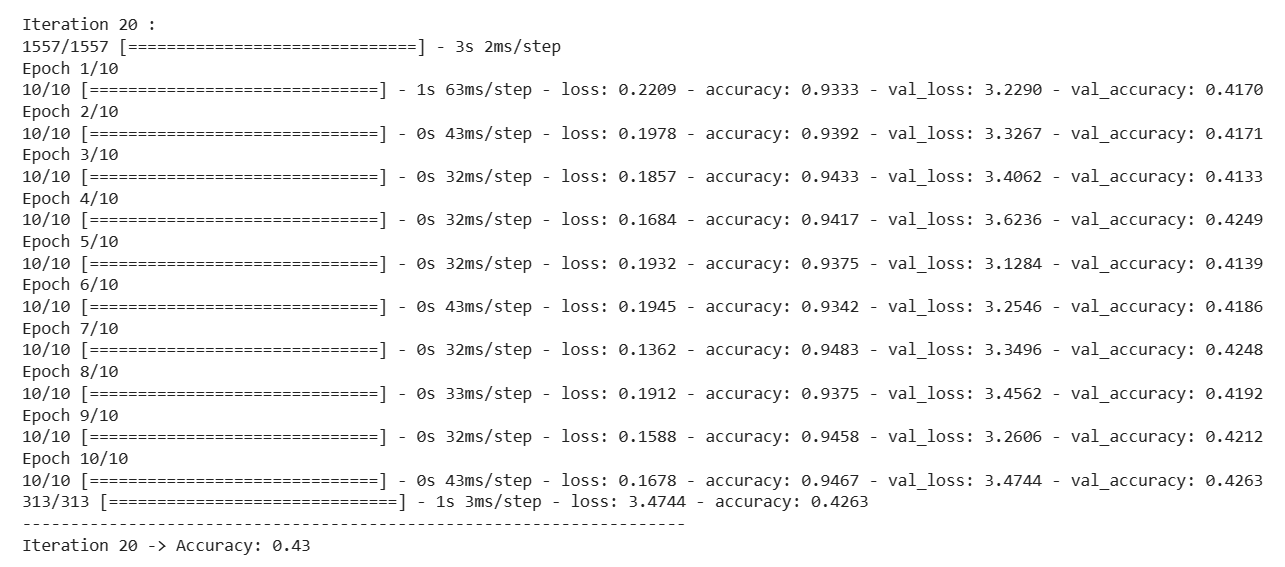
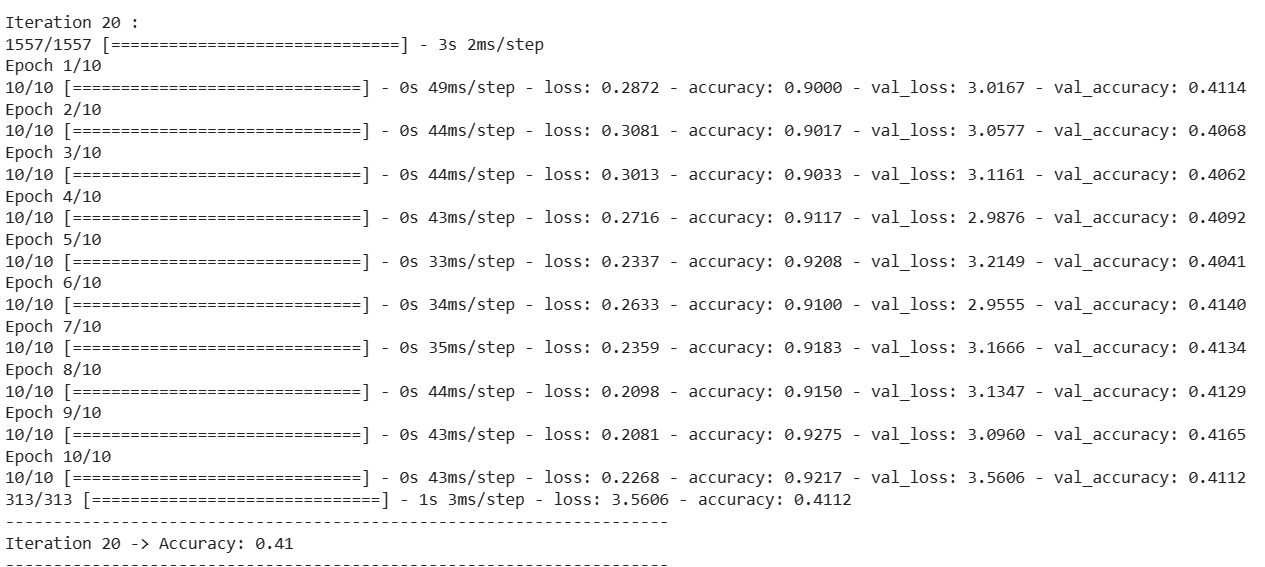
**So, overall accuracy on MNIST dataset after 20 iterations on this model:**

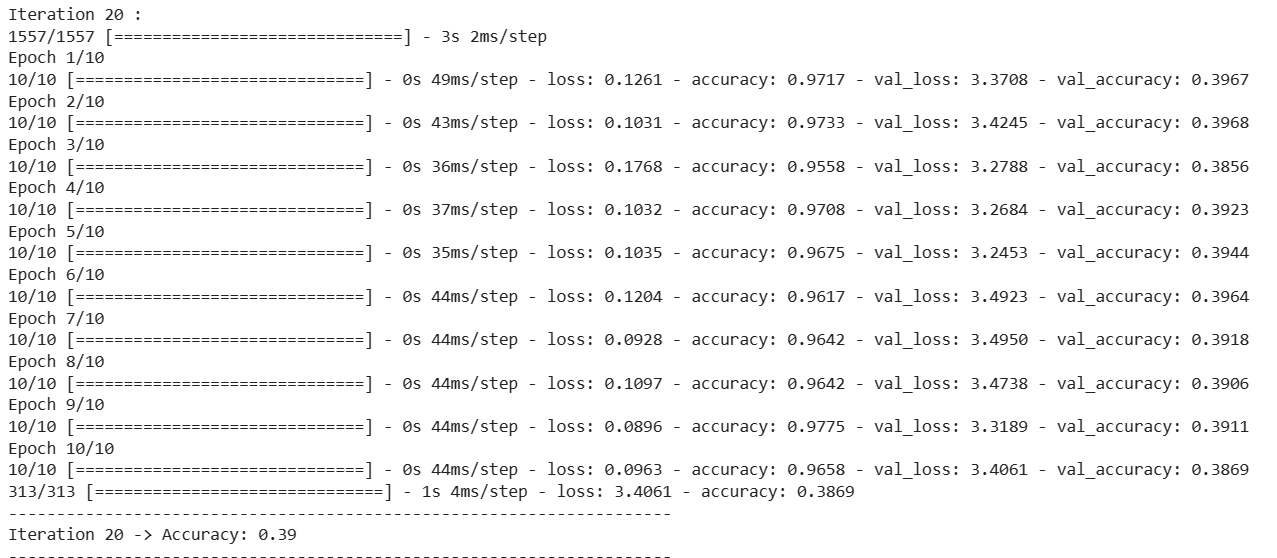


**Conclusion:**

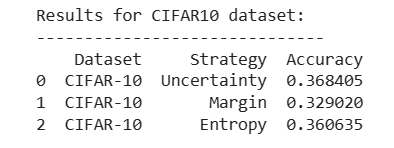
We conclude that the accuracy of the strategies are nearly the same or there is a small difference between them and they are all good and more than 95% with this model.

**Accuracy of CIFAR10 dataset after 20 iterations on this model with the 3 strategies:**

* with Uncertainty Strategy
* with Margin Strategy
* with Entropy Strategy



**So, overall accuracy of CIFAR10 dataset on 20 iterations on this model:**

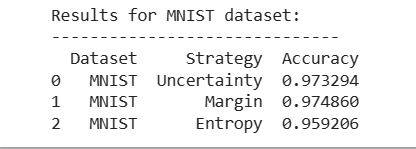


**Conclusion:**

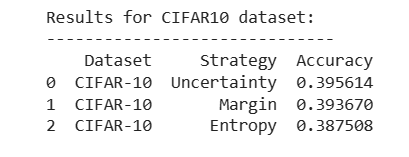
We conclude that the accuracy of the strategies is nearly the same and they are all completely less than the other dataset with a noticeable difference.

**We will make the same experiment but after 50 iterations on both datasets on the same model.**

* **On MNIST:**



* **On CIFAR10:**



**Conclusion:**

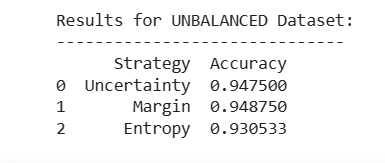
So, we conclude that the results are almost the same of all strategies, and they are all good except the results of CIFAR10 dataset which is totally bad and much less than the mnist.

**Experimental Analysis with Unbalanced Dataset**

We use the same analysis with the same structure of code but with an unbalanced dataset using make classification function from the sklearn.datasets.

We create a dataset with 10000 samples and 2 classes with weights [0.7, 0.3], which means that the first class will have 7000 samples, the second class will have 3000 samples.

**Then, the accuracy of the Unbalanced dataset after 30 iterations with different strategies:**



**Conclusion:**

The best results came from margin sampling, and accuracy of the strategies are nearly the same or there is a small difference between them and they all more than 90% with the dataset used.

**Experimental Analysis with a deep NLP model**

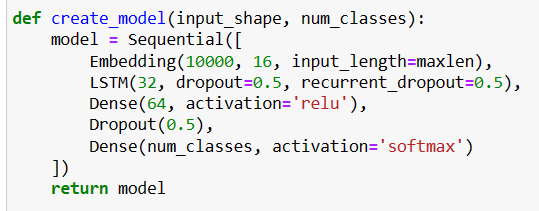
**We use the same analysis with the same structure of code but:**

Load and preprocess an NLP dataset instead of the MNIST or CIFAR-10 datasets, we use IMDB dataset which contains movie reviews labeled as positive or negative.

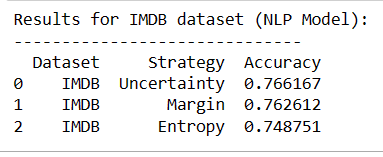
Instead of Conv2D and MaxPooling2D layers, we use layers like Embedding, LSTM, and Dense layers for our NLP model.

We update the uncertainty\_sampling, margin\_sampling, and entropy\_sampling functions to work with the output of our NLP model, which will be a probability distribution over the two classes (positive and negative).

**The model we use:**



**Accuracy of the NLP model after 30 iterations with the different strategies:**



**Conclusion:**

The accuracy of the strategies are nearly the same or there is a small difference between them and they all more than 75%.