# COL341 Spring 2023
# Make Up Assignment: Neural Network Regression
# (To be done Individually)

Due Date: 12th May 2023, Friday, 11:55PM (Hard deadline)
Demo Date: 13th May 2023, Saturday (Fixed)

## 1 Objective

The objective of this assignment is to help you understand the regression analysis process using neural networks. You will be required to build a regression model using Micro-Suturing Images and labels, which indicate the rating score by experienced doctors for micro-suturing.

## 2 Dataset

The micro-suturing dataset (present in this link) contains Suturing Images and rating scores by experienced doctors. The rating score is given on a scale of 1 to 9, where 1 represents a poorly executed suture, and 9 represents a perfectly executed suture.

## 3 TASK 1

Your task is to build a regression model using the VGG-19 model and fine-tune it on the Suturing Image Dataset.

### 3.1 Model Architecture

You will be using the VGG-19 model as a starting point for your regression model. VGG-19 is a deep convolutional neural network with 19 layers, pre-trained on the ImageNet dataset. You will fine-tune the VGG-19 model by replacing the last fully connected layer with a new layer that has one output unit for regression. You will also freeze the weights of the convolutional layers and only train the weights of the newly added layer.

## 3.2 Performance Metrics

You will evaluate the performance of your regression models on the Suturing Image Dataset using the Mean Squared Error (MSE) and the R-squared score. MSE is a common metric used for regression tasks that measures the average squared differences between the predicted and actual values. R-squared score measures the proportion of variance in the dependent variable that is predictable from the independent variables.

## 3.3 Ablation Study

You will perform multiple ablations on the fine-tuned VGG-19 model to understand the effect of different activation functions, additional layers, and changes in the learning rate.

1. Experiment for each change is to be performed to show the effect of using batch norm, dropout, and freezing weights for different numbers of layers.

2. Experiment where you need to change activation functions such as ReLU, LeakyReLU, Sigmoid, and Tanh.

3. Experiment showing effect in model performance with respect to change in learning rate.

You are required to report metrics and loss curves with respect to each of the experiments.

# 4 TASK 2: Interpret the predictions

In this part of the assignment, you should explore methods to interpret the predictions of their finetuned VGG network. They can use techniques such as:

1. Feature Visualization: This technique can be used to generate images that maximize the activation of a specific neuron or layer in the VGG network. The students can use PyTorch's built-in hooks and *torch.autograd* to generate these visualizations.

2. Attribution Methods: These methods can be used to identify the regions of an image that the model is using to make its predictions. Some popular attribution methods include $Grad{-}CAM\ and\ LRP(Layerwise\ Relevance\ Propagation)$. The students can use third-party libraries like *captum* to generate these attributions.

3. Analyze the results: Finally, the students should analyze the results of their interpretation and should identify any patterns.

## 4.1 Related resources

1. Grad CAM
2. Filter visualization
3. Miscellaneous

# 5 TASK 3: Improving the Regression Model

In the first part of the assignment, you built a regression model using the VGG-19 model and fine-tuned it on the Suturing Image Dataset. In this part of the assignment, you will have the opportunity to make changes to the network architecture and train a new model to try and improve its performance.

Your task is to experiment with different network architectures and hyper-parameters to improve the performance of the regression model. You can try out different variations of the VGG-19 model or even try out different models altogether. Additionally, you can experiment with different learning rates, optimizers, and regularization techniques. Your goal is to achieve a lower mean squared error (MSE) and a higher R-squared score than the model you built in the first part of the assignment.

## 5.1 Guidelines

Here are some guidelines to help you get started:

1. Experiment with different network architectures: You can try variations of the VGG-19 model, such as VGG-16 or ResNet. You can also try building your own custom architecture, perhaps with more or fewer layers or with different types of layers.
2. Try different activation functions: You can experiment with different activation functions such as ReLU, LeakyReLU, Sigmoid, and Tanh. You can also try out some newer activation functions like Swish or GELU.
3. Experiment with different optimizers: You can try out different optimizers such as SGD, Adam, and RMSprop. Each optimizer has its own strengths and weaknesses, and you should experiment to find the best one for your model.
4. Use regularization techniques: Regularization techniques such as L1/L2 regularization, dropout, and batch normalization can help prevent over-fitting and improve the generalization ability of your model.

# 6 Submission

You are required to submit a report that includes the following:

1. Detailed description of the model architecture and the fine-tuning process
2. Results and analysis of the different ablation experiments
3. Comparative analysis of the performances of different models with loss curves and related curves

Codes for all the ablations and models need to be submitted. Model weights should be uploaded to Onedrive/ Google Drive and need to be submitted through link sharing. Mention the links to model weights in report. You are required to make a test script where you read images and their labels from a given folder and save R-squared score and Mean Squared Error for each of the ablation/experiment model in a csv format.

```
python test.py \
    --model_path <path_to_model> \
    --input_path <path_to_data_folder> \
    --output_path <path_to_output_csv>
```

You will be evaluated on a hidden test set at the time of the demo. Test data will be in the same format as the training data i.e. contains one annotation file and an image folder.