MACHINE LEARNING C
Southern California Trial Rules

1. **DESCRIPTION:** Teams will develop computational models to accomplish tasks and complete a written test on machine learning theory and applications.

   A TEAM OF UP TO: 2 **EVENT TIME:** 50 minutes

2. **EVENT PARAMETERS:**
   a. Each team may bring up to two 8.5” x 11” sheets of paper, which may be in a sheet protector sealed by tape or laminated, that may contain information on both sides in any form and from any source without any annotations or labels affixed.
   b. Each team may also bring writing utensils and two stand-alone non-programmable, non-graphing calculators. Teams will not have access to the internet during the event.

3. **THE COMPETITION:**

   Part I: Pre-Tournament Task
   a. Participants aim to predict labels from features using a computational model. Tasks, provided by the Event Supervisor, will consist of a Training Dataset and a Test Dataset. The Training Dataset consists of features and labels, while the Test Dataset only contains features. The performance of each model will be evaluated based on a given metric using an unknown subset of the Test Dataset.
   b. The Event Supervisor must release the Task at least two weeks prior to the competition. Online submissions, through Kaggle Classroom, are accepted until the start of the Tournament’s first event.
   c. Tasks are limited to the following categories: binary or multi-class classification, two-dimensional object detection, and regression of a single variable.
   d. Participants must run their code in a Kaggle Kernel to generate a formatted CSV file and submit the CSV to the Kaggle competition. The Kernel must be published and shared with the Event Supervisor.

   Part II: Written Test

   Teams will be given 50 minutes to complete a written test on the topics below, with questions including but not limited to: writing pseudocode, describing common algorithms, explaining concepts, performing calculations, and applying machine learning concepts and algorithms to real-world problems.
   a. Mathematics: regular, joint, and conditional probabilities; Bayes’ Theorem; understanding of random variables; cross-entropy; Kullback-Leibler Divergence; evidence-lower bound
   b. Optimization: gradient descent (GD); variations of GD including momentum, Nesterov momentum, Adam, AdaProp, RMSProp; backpropagation algorithm; difficulties training neural networks
   c. Convolutional architecture design principles: architectures of famous ImageNet models and what made them successful (AlexNet, ResNet, VGG, GoogLe Net); types of layers in neural networks and their theory and applications (max pooling, dense layers, flattening, bottleneck, nonlinearities, etc.)
   d. Evaluation and visualization of deep learning results: precision-recall curves; t-SNE, UMAP, PCA; confusion matrix; training and validation curves; saliency maps; gradient ascent
   e. Implementation of deep learning methods: characteristics of ML libraries including tensorflow, PyTorch; static vs. dynamic graphs; characteristics of processors including CPUs, GPUs, TPUs
   f. Computer vision: common tasks and models used to solve them; datasets used to evaluate these models
   g. Natural language processing: word embeddings, including Bag of Words & word vectors; transformer architectures, including BERT & GPT; sequential neural networks and recurrent neural networks
   h. Generative models: generative adversarial network architecture and applications; variational autoencoder architecture and applications

4. **SCORING:**
   a. High score wins; Final Score = (TS + 1) x (ES + 0.2).
   b. Task Score (TS) = (Part I score / Highest Part I score for all teams). TS = 0 if any result is hard coded.
   c. Exam Score (ES) = (Part II score / Highest Part II score for all teams)
   d. Ties will be broken by 1) Task Score 2) selected questions from the written test.

RECOMMENDED RESOURCES: pytorch.org, tensorflow.org, python.org, kaggle.com