# ECE523: Engineering Applications of Machine Learning and Data Analytics

I acknowledge that this exam is solely my effort. I have done this work by myself. I have not consulted with others about this exam in any way. I have not received outside aid (outside of my own brain) on this exam. I understand that violation of these rules contradicts the class policy on academic integrity.

**Name**: ______________________________

**Signature**: ______________________________

**Date**: ______________________________

**Instructions**: There are four problems. You have 50 minutes to complete the exam. Partial credit is given for answers that are partially correct. No credit is given for answers that are wrong or illegible. Write neatly.

Problem 1: __________

Problem 2: __________

Problem 3: __________

Problem 4: __________

Total: __________

# Problem #1 – Backpropagation (10 Points)

In class, we discussed the backpropagation algorithm and derived a methodology to be able to update the weights of an artificial neural network. Describe the backpropagation algorithm in some detail. Do not just write the update equation! You should begin by starting out with the objective you're trying to optimize and why we needed to exploit rules from calculus I be able to find these updates.

## Problem #1 – Backpropagation (cnt'd)

# Problem #2 – Neural Nets with Regularization (10 Points)

One method for preventing the neural networks' weights from overfitting is to add regularization terms. You will now derive the update rules for the regularized neural network. Recall that the non-regularized gradient descent update rule for $w_1^{t+1}$ is:

$$w_{ji}^{t+1} = w_{ji}^{t} + \eta \sum_{n=1}^{N} e_j(n)\phi'\left(v_j(n)\right) y_i(n) \tag{1}$$

Derive the update rule for $w_{ji}^{t+1}$ in the regularized neural net loss function which penalizes based on the square of each weight. Use $\lambda \geq 0$ to denote the regularization parameter. Use the following regularizer:

$$R(w) = \lambda \sum_i w_i^2$$

**Bonus (2pts)**: Re-express the regularized update rule so that the only difference between the regularized setting and the unregularized setting above is that the old weight $w_{ji}^t$ is scaled by some constant. Explain how this scaling prevents overfitting.

# Problem #3 – Random Short Answer (15 Points)

(SA:1) Describe the process of learning and testing a random forest on a data set with $n$ samples and $p$ features.

(SA:2) What is an appropriate way to train a deep neural network? The key word in that sentence is "appropriate".

(SA:3) What does a shatter coefficient mean in the context of risk and VC-dimension (Vapnik-Chervonenkis)? What is the VC-dimension of a linear classifier trained on a data set with $p$ dimensions.

(SA:4) In semi-supervised learning, we discussed self- and co-training. Briefly describe these two semi-supervised learning techniques. Are both techniques always applicable?

# Problem #4 – True/False: A Gamblers Ruin (10 Points)

**[True/False] (1 point)**: If $f \in \mathcal{F}$ is function in a class of functions and $N = |\mathcal{F}|$ then the error for all $\mathcal{F}$ is upper bounded by:

$$\mathbb{P}\left(\left|\frac{1}{n}\sum_{i=1}^{n} Z_i - p\right| \geq \epsilon\right) \leq 2\exp\left(-2n\epsilon^2\right)$$

where $Z_i$ is a Bernoulli random variable indicating an error and $\epsilon > 0$.

**[True/False] (1 point)**: Passive-aggressive online learning updates a linear model at each time step $t$ regardless if a mistake was made or not.

**[True/False] (1 point)**: The multi-armed bandit addresses problems that require exploration of new arms and exploitation of the ones we know perform well.

**[True/False] (1 point)**: The Vapnik-Chervonenkis dimension allowed use to bound a classifier's error of a class of functions $\mathcal{F}$ without needed to worry about $N = |\mathcal{F}|$ being extremely large or possibly infinite.

**[True/False] (1 point)**: One of the disadvantages of deep learning with auto-encoders is that we need a large volume of labeled data to train each layer.

**[True/False] (1 point)**: In the context of a adversarial MAB, the term $\gamma \in [0, 1]$ controls the trade-off between the estimated reward of the arm and pure exploration.

$$\widehat{p}_i(t) = \gamma \frac{w_i(t)}{\sum_j w_j(t)} + (1-\gamma)\frac{1}{K}$$

where $K$ is the number of arms and $w_i(t)$ is the weight of the $i$th arm at time $t$.

**[True/False] (1 point)**: A neural network will (likely) find a local minimum for its optimization problem and the same is true for a support vector machine.

**[True/False] (1 point)**: Using a sigmoid activation function in a neural network trained with backpropagation is one way to avoid the vanishing gradient problem.

**[True/False] (0 point)**: To the student who answered False last time on the question about free points: No more free points!

**[True/False] (1 point)**: A discriminator network, $D$, after enough training in a GAN will always be able to identify if a sample came from the data set or the generator network, $G$.

**[True/False] (1 point)**: In backpropagation, the only difference between updating a hidden node versus an output node is how the local gradient is calculated.

# Scratch Paper (not graded)