1. (6 points) **True or False**
   
   (a) (0.5 points) Consider a cancer diagnosis classification problem where almost all of the people being diagnosed don’t have cancer. The probability of correct classification is the most important metric to optimize.

   A. True  
   B. False 

   (b) (0.5 points) The L2 penalty in a ridge regression is equivalent to a Laplace prior on the weights.

   A. True  
   B. False 

   (c) (0.5 points) The log-likelihood of the data will always increase through successive iterations of the expectation maximization algorithm.

   A. True  
   B. False
(d) (0.5 points) K-means automatically adjusts the number of clusters.

A. True  B. False

(e) (0.5 points) The largest eigenvector of the covariance matrix is the direction of minimum variance in the data.

A. True  B. False

(f) (0.5 points) The Naive Bayes classifier uses the maximum a posteriori or the MAP decision rule for classification.

A. True  B. False

(g) (0.5 points) In SVMs, the values of $\alpha_i$ for non-support vectors are 0.

A. True  B. False

(h) (0.5 points) The depth of a learned decision tree can be larger than the number of training examples used to create the tree.

A. True  B. False

(i) (0.5 points) Any multi-layered neural network with linear activation functions at hidden layers can be represented as a neural network without any hidden layer.

A. True  B. False

(j) (0.5 points) Suppose a convolutional neural network is trained on ImageNet dataset (Object recognition). This trained model is then given a completely white image as an input. The output probabilities for this input would be equal for all classes.

A. True  B. False

(k) (0.5 points) Grid hyperparameter search usually preferred over random search.

A. True  B. False

(l) (0.5 points) Adagrad has different per-element stepsizes. This allows parameters with large gradients to change more quickly than those with large step-sizes.

A. True  B. False
2. (6 points) **Multiple Choice**

(a) (0.5 points) Which of the following methods can achieve zero training error on any linearly separable dataset?

- A. Decision tree.
- B. Hard-margin SVM.
- C. 15-nearest neighbors.
- D. Neural Network.

(b) (0.5 points) Jonathan just trained a decision tree for a digit recognition. He notices an extremely low training error, but an abnormally large test error. He also notices that an SVM with a linear kernel performs much better than his tree. What could be the cause of his problem?

- A. Decision tree is too deep.
- B. Learning rate too high.
- C. A Decision tree is overfitting.
- D. There is too much training data.

(c) (0.5 points) Jonathan has now switched to multilayer neural networks and notices that the training error is going down and converges to a local minimum. Then when he tests on the new data, the test error is abnormally high. What is probably going wrong and what do you recommend him to do?

- A. Training data size is not large enough, collect more data and retrain it.
- B. Play with learning rate and add regularization to the loss function.
- C. Use a different initialization and train the network several times, use the average of predictions from all nets to predict test data.
- D. Use the same training data but add two more hidden layers.

(d) (0.5 points) Lasso can be interpreted as least-squares linear regression where

- A. weights are regularized with the L1 norm.
- B. the weights have a Gaussian prior.
- C. weights are regularized with the L2 norm.
- D. the solution algorithm is simpler.

(e) (0.5 points) Why is PCA sometimes used as a preprocessing step before regression?

- A. To reduce overfitting by removing poorly predictive dimensions.
- B. To make computation faster by reducing the dimensionality of the data.
- C. To expose information missing from the input data.
- D. For inference, we prefer features that are not axis-aligned.

(f) (0.5 points) In a soft-margin support vector machine, decreasing the slack penalty term \( C \) causes

- A. more overfitting.
- B. a smaller margin.
- C. less overfitting.
- D. less sensitivity to outliers.
(g) (0.5 points) In which neural net architecture, does weight sharing occur?
   A. Convolutional neural Network.
   B. Recurrent Neural Network.
   C. Fully Connected Neural Network.
   D. None of the above.

(h) (0.5 points) Which of the following is/are forms of regularization in neural networks.
   A. Weight decay.
   B. L1 regularization.
   C. Batch Normalization.
   D. Dropout.

(i) (0.5 points) You are training a neural network, but the training error is high. Which of the following, if done in isolation, has a better-than-tiny chance of reducing the training error?
   A. Adding another hidden layer.
   B. Normalizing the input data.
   C. Adding more units to hidden layers.
   D. Training on more data.

(j) (0.5 points) Which of the following statements is true when you use 1x1 convolutions in a CNN?
   A. It can help in dimensionality reduction.
   B. It can be used for feature pooling.
   C. It suffers less overfitting due to small kernel size.
   D. None of the above.

(k) (0.5 points) Batch Normalization is helpful because?
   A. It normalizes (changes) all the input before sending it to the next layer.
   B. It returns back the normalized mean and standard deviation of weights.
   C. It is a very efficient backpropagation technique.
   D. None of the above.

(l) (0.5 points) Consider one layer of weights in a convolutional neural network for grayscale images, connecting one layer of units to the next layer of units. Which type of layer has the fewest parameters to be learned during training? (Select one.)
   A. A convolutional layer with 10 3 x 3 filters.
   B. A max-pooling layer that reduces a 10 x 10 image to 5 x 5.
   C. A convolutional layer with 8 5 x 5 filters.
   D. A fully-connected layer from 20 hidden units to 4 output units.
3. (6 points) Short Answers

(a) (1 point) Explain what effect will the following operations generally have on the bias and variance of your model. Fill in one of 'increases', 'decreases' or 'no change' in each of the cells:

<table>
<thead>
<tr>
<th>Operation</th>
<th>Bias</th>
<th>Variance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regularizing the weights</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Increasing the size of the layers</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(more hidden units per layer)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Using dropout to train a deep neural network</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Getting more training data</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(from the same distribution as before)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Consider the following two multilayer perceptrons, where all of the layers use linear activation functions.

(b) (0.5 points) Give one advantage of Network A over Network B.

(c) (0.5 points) Give one advantage of Network B over Network A.
Suppose you want to redesign the AlexNet architecture to reduce the number of arithmetic operations required for each backprop update.

(d) (0.5 points) Would you try to cut down on the number of weights, units, or connections? Justify your answer.

(e) (0.5 points) Would you modify the convolution layers or the fully connected layers? Justify your answer.

(f) (0.5 points) Suppose you have a convolutional network with following architecture:
- The input is an RGB image of size 256 x 256.
- The first layer is a convolution layer with 32 feature maps and filters of size 3 x 3. It uses a stride of 1, so it has the same width and height as original image.
- The next layer is a pooling layer with a stride of 2 (so it reduces the size of each dimension by a factor of 2) and pooling groups of size 3 x 3.

Determine the size of the receptive field for a single unit in the pooling layer. (I.e., determine the size of the region of the input image which influences the activation of that unit.) You may assume the receptive field lies entirely within the image. Hint: you may want to draw a one-dimensional conv net to reason about this problem.
Consider the graph in Figure 1 representing the training procedure of a GAN:

Figure 1: Cost function of the generator plotted against the output of the discriminator when given a generated image $G(z)$. Concerning the discriminator’s output, we consider that 0 (resp. 1) means that the discriminator thinks the input ”has been generated by G” (resp. ”comes from the real data”).

(g) (0.5 points) Early in the training, is the value of $D(G(z))$ closer to 0 or closer to 1? Explain.

(h) (0.5 points) Two cost functions are presented in figure (1), which one would you choose to train your GAN? Justify your answer.

(i) (0.5 points) You know that your GAN is trained when $D(G(z))$ is close to 1. True / False? Explain.
(j) (0.5 points) Determine what the following recurrent network computes. More precisely, determine the function computed by the output unit at the final time step; the other outputs are not important. All of the biases are 0. You may assume the inputs are integer valued and the length of the input sequence is even.

(k) (0.5 points) Recall the LSTM architecture. Suppose you want the memory cell to sum its inputs over time. What values should the input gate and forget gate take? You do not need to justify your answer.
4. (6 points) **MLE, Linear Regression and Bayes Net**

(a) (1 point) You are given a collection of \(n\) documents, where the word count of the \(i\)-th document is \(x_i\). Assume that the word count is given by an Exponential Distribution with parameter \(\lambda\). In other words, for a non-negative integer \(x\),

\[
P(wordcount = x|\lambda) = \frac{\lambda^x \exp(-\lambda)}{x!}
\]

Compute \(\lambda\) such that the likelihood of observing \(\{x_1, x_2, ..., x_n\}\) is maximized.

Consider a linear regression problem \(y = w_1 x + w_0\), with a training set having \(m\) examples \((x_1, y_1), \ldots, (x_m, y_m)\). Suppose that we wish to minimize the mean fifth degree error (loss function) given by:

\[
Loss = \frac{1}{m} \sum_{i=1}^{m} (y_i - w_1 x_i - w_0)^5
\]
(b) (0.5 points) Calculate the gradient with respect to the parameters $w_1$ and $w_0$. 

(c) (1 point) Write down pseudo-code for online gradient descent for this problem. 

(d) (0.5 points) Give one reason in favor of online gradient descent compared to batch gradient descent, and one reason in favor of batch over online.
Consider a Bayesian network $B$ with boolean variables.

(e) (0.5 points) Is there any variable(s) conditionally independent of $X_{33}$ given $X_{11}$ and $X_{12}$? If so, list all.

(f) (0.5 points) Is there any variable(s) conditionally independent of $X_{33}$ given $X_{22}$? If so, list all.

(g) (0.5 points) Write the joint probability $P(X_{11}; X_{12}; X_{13}; X_{21}; X_{22}; X_{31}; X_{32}; X_{33})$ factored according to the Bayes net. How many parameters are necessary to define the conditional probability distributions for this Bayesian network?
(h) (0.5 points) Write an expression for $P(X_{13} = 0; X_{22} = 1; X_{33} = 0)$ in terms of the conditional probability distributions given in your answer to part (g).

(i) (0.5 points) From your answer to (h), can you say $X_{13}$ and $X_{33}$ are independent? Why?

(j) (0.5 points) Can you say the same thing when $X_{22} = 1$? In other words, can you say $X_{13}$ and $X_{33}$ are independent given $X_{22} = 1$? Why?

5. (6 points) **Decision Trees and Neural Networks**

Consider the training dataset given below. In the dataset, X1, X2, and X3 are the attributes and Y is the class variable.

<table>
<thead>
<tr>
<th>Example#</th>
<th>$X_1$</th>
<th>$X_2$</th>
<th>$X_3$</th>
<th>$Y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>E1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>+ve</td>
</tr>
<tr>
<td>E2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>-ve</td>
</tr>
<tr>
<td>E3</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>-ve</td>
</tr>
<tr>
<td>E4</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>+ve</td>
</tr>
<tr>
<td>E5</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>-ve</td>
</tr>
</tbody>
</table>
(a) (0.5 points) Which attribute has the highest information gain? Justify answer?

(b) (1 point) Draw the decision tree for this dataset using the information gain criteria.

(c) (0.5 points) Suppose that we have the following validation dataset:

<table>
<thead>
<tr>
<th>Example#</th>
<th>$X_1$</th>
<th>$X_2$</th>
<th>$X_3$</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>E6</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>+ve</td>
</tr>
<tr>
<td>E7</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>-ve</td>
</tr>
<tr>
<td>E8</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>-ve</td>
</tr>
</tbody>
</table>

Can you construct a decision tree that will have 100% accuracy on the validation dataset as well as the training dataset. Answer Yes or No. If your answer is yes, then draw the decision tree that will have 100% accuracy. If your answer is no, then explain why it cannot be done.
Consider the neural network given below:

Assume that all internal nodes compute the sigmoid function. Write an explicit expression to how back propagation (applied to minimize the least squares error function) changes the values of $w_1, w_2, w_3, w_4$ and $w_5$ when the algorithm is given the example $x_1 = 0, x_2 = 1$, with the desired response $y = 0$ (notice that $x_0 = 1$ is the bias term). Assume that the learning rate is $\alpha$ and that the current values of the weights are: $w_1 = 3, w_2 = 2, w_3 = 2, w_4 = 3$ and $w_5 = 2$. Let $o_j$ be the output of the hidden units and output units indexed by $j$. 
(d) (0.5 points) Forward propagation. Write equations for \( o_1, o_2 \) and \( o_3 \) in terms of the given weights and example.

(e) (0.5 points) Backward propagation. Write equations for \( \delta_1, \delta_2 \) and \( \delta_3 \) in terms of the given weights and example.

(f) (1 point) Give an explicit expression for the new weights.

(g) (2 points) One of the interesting features of the ReLU activation function is that it sparsifies the activations and the derivatives, i.e. sets a large fraction of the values to zero for any given input vector. Consider the following network:
Note that each $w_i$ refers to the weight on a single connection, not the whole layer. Suppose we are trying to minimize a loss function $L$ which depends only on the activation of the output unit $y$. (For instance, $L$ could be the squared error loss $\frac{1}{2}(y-t)^2$.) Suppose the unit $h_1$ receives an input of -1 on a particular training case, so the ReLU evaluates to 0. Based only on this information, which of the weight derivatives $\frac{\partial L}{\partial W_1}, \frac{\partial L}{\partial W_2}, \frac{\partial L}{\partial W_3}$ are guaranteed to be 0 for this training case? Write YES or NO for each. Justify your answers.
6. (2 points (bonus)) For the image below, highlight two modules that could leverage pre-trained models. What pretrained models could you use?

Which of the following systems use only deep network components?

A. R-CNN.
B. Fast R-CNN.
C. Faster R-CNN.
D. None of the above.
[scratch paper]