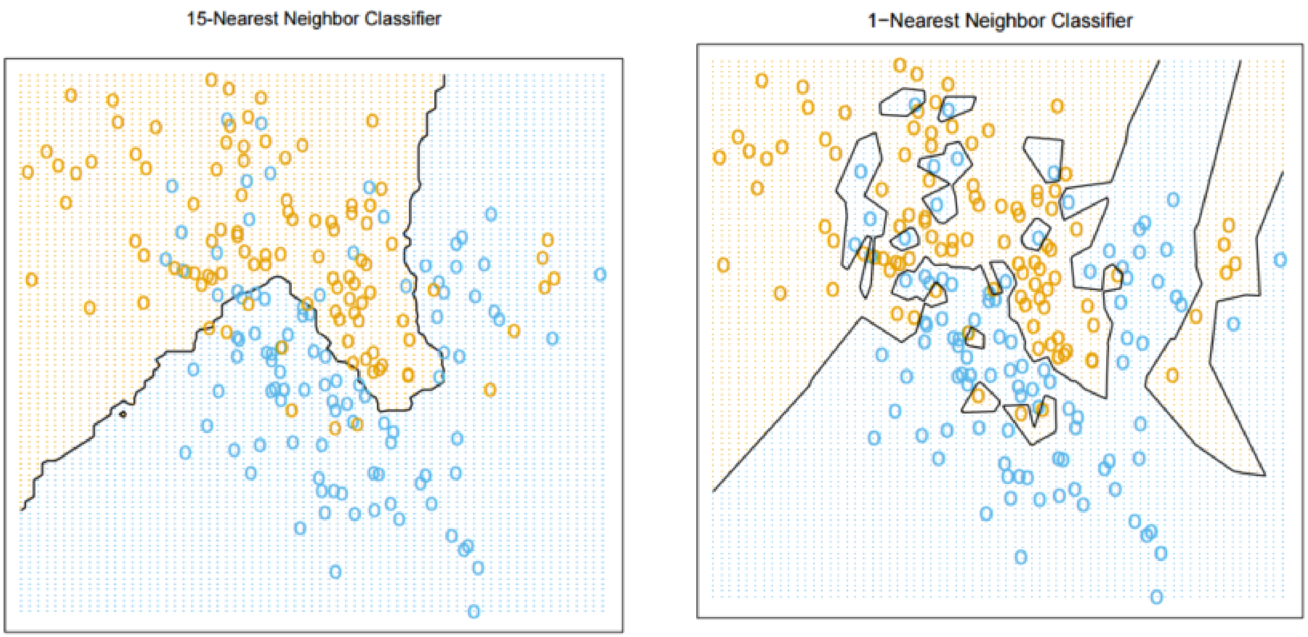
1. Suppose our training set and test set are the same. Why would this be a problem?
2. Why is it necessary to have both a test set and a validation set?
3. Images are generally represented as $n \times m \times 3$ arrays, but we treat them as vectors. How is that possible?
4. Write pseudocode to select the k in the k-nearest-neighbours algorithm.
5. Give an example of a training and a test set such that 3-nearest-neighbours and 5-nearest neighbours perform differently on the test set
6. Consider the data plotted below. If all that you have to go by is the data below, how would you objectively determine whether 1-NN or 15-NN is the better classifier for the data?



7. What is the performance of k-NN on the training set for k = 1? Is it the same for k = 3? (Give an example in your answer to the second question)
8. What happens to the performance of k-NN if k is the same as the size of the training set?
9. Explain how the quadratic cost function

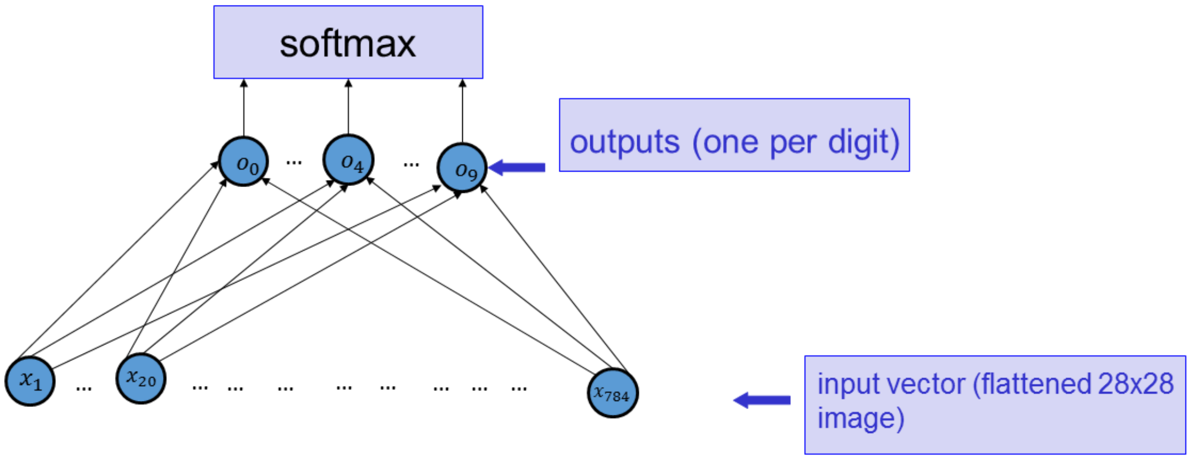
$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$$

is constructed. Specifically, what is the intuitive meaning of $(h_\theta(x^{(i)}) - y^{(i)})^2$?

10. How to estimate the best h_θ by looking at a surface plot of the cost function?
11. How to estimate the best h_θ by looking at a contour plot of the cost function?
12. Write a Python function to find a local minimum of $y = x^6 - 10x^5 - 5x^4 + 1$ using gradient descent
13. On a plot of a function of one variable $y = f(x)$, at point $(x_0, f(x_0))$, write down a unit vector that points "uphill."

14. Explain how to transform a two-class classification problem into a regression problem. Why won't this approach work for multi-class classification?
15. Write the pseudocode for training and evaluating a one-vs-all classifier based on linear regression
16. What is the decision boundary for a linear-regression based classifier with multiple predictor variables/features?
17. Derive the average squared-differences cost function for linear regression by assuming that the data is generated using
$$y = \theta_0 + \theta_1 x + \epsilon, \epsilon \sim N(0, \sigma^2).$$
18. Derive the cost function for that corresponds to the likelihood of the data in Poisson regression. In Poisson regression, we assume that
$$y \sim Poisson(\theta^T x).$$
(Reminder: that means that $P_\lambda(y = k) = \frac{\lambda^k \exp(-\lambda)}{k!}$ for $\lambda = \theta^T x$.)
19. Derive the gradient of the cost function in Problem 18.
20. Write Python code to perform Poisson regression using gradient descent.
21. Assuming you found the best θ using the code from Problem 20, how would you use it to predict the y for a new x?
22. Suppose that your model is
$$y \sim Poisson(\alpha x),$$
with $\alpha \in \mathbb{R}$. Assume you believe that $\alpha \sim N(0, \beta)$. Write Python code to obtain the posterior distribution of α given a dataset, and to obtain the MAP estimate of α.
23. Make up a neural network that uses neurons with the logistic function activation that computes the logical NOR.
24. Write vectorized code (i.e., you cannot use for-loops) to compute the output of a feedforward neural network
25. What is the disadvatange of using a single output when using feedforward neural networks for classification?
26. What is the disadvatange of using the quadratic cost function when using feedforward neural networks for classification?
27. What is the partial derivative of the quadratic cost function for a feedforward neural network with respect to a bias for a neuron in the output layer?
28. Vectorize the computation on Slide 6 of `http://www.cs.toronto.edu/~guerzhoy/321/lec/W04/backprop.pdf`
29. Explain why the output of Softmax is better for representing probabilities than the raw output layer.
30. What is one advantage of the `tanh` activation function compared to the `sigmoid` activation function?
31. Explain what it means for a neuron to be dead. Why is that a bad thing? What is the implication for initializing the weights and biases for ReLU units?
32. Is `@ml_hipster` funny?

33. Consider the feedforward network below.



Explain why it does not matter much whether we use the identity activation function or the tanh activation function in the `outputs` layer.

34. Why is it useful to normalize the input data?

35. The learning curve for the training set is "wiggly" on Slide 3 of `http://www.cs.toronto.edu/~guerzhoy/321/lec/W05/overfitting.pdf`. Why? (There are two possible reasons.) How would you produce a smoother learning curve?

36. Devise of an example of fitting a curve in 2D where a cubic polynomial would overfit, but a quadratic polynomial will work. In your example, sketch the points in the training and test sets and provide a cost function.

37. Suppose we modify our cost function as follows:

$$cost_{wd} = cost + \frac{\lambda}{3}\sum_{i,j,k}(W^{(k,i,j)})^3.$$

How would you compute the gradient of $cost_{wd}$?

Why is adding the cubes of the weights to the cost function a bad idea?

38. Describe two different scenarios in which you would not observe overfitting

39. Prove that the derivative of the negative log-probability of the right answer cost function with respect to the layer below the softmax in a network on slide 6 of `http://www.cs.toronto.edu/~guerzhoy/321/lec/W04/onehot.pdf` is

$$\frac{\partial C}{\partial o_i} = p_i - y_i.$$

40. Write code to generate a synthentic dataset for which the weights in a neural network on slide 7 of `http://www.cs.toronto.edu/~guerzhoy/321/lec/W05/overfitting_prevent.pdf` would look differently (approximately as described on slide 7) under L1 and L2 norm regularization.
41. Don't you feel sorry for life science students who have to memorize the stuff about dendrites and axons?
42. how is the replicated feature approach related to the invariant feature approach?
43. On slide 5 fo `http://www.cs.toronto.edu/~guerzhoy/321/lec/W06/convnet_intro.pdf`, you see a way of obtaining an image where points near horizontal edges are bright and everything else is dark, but this only works for edges that happen across a step of 2 pixels (or a little bit more). How would you produce an image that shows horizontal edges where the transition between one area and another is more gradual, and happens over, say, 10 pixels?
44. Explain Slide 15 of `http://www.cs.toronto.edu/~guerzhoy/321/lec/W06/convnet_intro.pdf`: why is $(N - F)/stride + 1$ the output size? (What do we mean by "output size?" Precisely define what N, F, and $stride$ refer to.)
45. Why to we sometimes pad the border of the image with zeros when performing convolutions?
46. Explain the difference between Max Pooling and Average Pooling. Write Python code that performs both.
47. When might you expect Max Pooling to work bettern (and vice-verse)?
48. Give an example of a gradient computation when a network has a convolutional layer.
49. Give an example of a gradient computation when a network has a max-pooling layer.
50. Give an example of a gradient computation when a network has both a convolutional layer and a max-pooling layer.
51. Suppose the size of an input layer is $32 \times 32 \times 3$, and you have 10 5×5 filters with stride 2 and pad 2. What is the number of parameters (weights+biases) that we need to define the connections between the layers? What is the size of the output layer?
52. Why would we use 1×1 convolutions?
53. What is the idea behind the Inception module?
54. How many activation functions need to be computed if we are computing the first convolutional layer of a network which takes as input an $N \times N \times 3$ image using M 3×3 features, using 0 padding and a stride of 1?
55. Suppose we want to visualize what a neuron in a ConvNet is doing. How would you go about that?
56. Sometimes when we visualize what a neuron is doing, we display an image that's smaller than an input image. How is that done?
57. How is what the neurons are doing in the lower layers (near the input) different from what the neurons are doing in the upper layers?
58. Explain guided backpropagation. Provide pseudocode, and explain how guided backprop improves on simply computing the gradient.
59. How does Deep Dream work? Provide pseudocode.
60. Explain the cost function for Neural Style Transfer, and explain how Neural Style Transfer works. Reminder: the Gram matrix is $G^l_{ij} = \sum_k F(y)^l_{ik} F(y)^l_{jk}$.

61. Explain the cost function for training RNNs
62. Explain the vanishing gradient problem
63. Explain how to do machine translation with LSTM. Sketch the network. What is the cost function?
64. Why do mini-batches need to be class-balanced?
65. Explain the momentum method. State the equations used to update the parameters
66. What is a simple way to run gradient descent with adaptive learning rates?
67. In Bayesian inference, we'd like to compute $P(\theta|data)$ for the training data. Explain how to do that, and relate your explanations to our concepts of the cost function and of regularization.
68. When doing full Bayesian inference, how would you make predictions for new data?
69. What is the basic idea of Monte Carlo methods in the context of full Bayesian inference?
70. Explain the Metropolis algorithm.
71. Explain the computation of $P(x)$ in RBMs in terms of features of the input and hypotheses about which of the features are present.
72. For an RBM, show that $P(h_j = 1|x) = \sigma(W_{j,:}x + b_j)$
73. Explain how autoencoder networks work
74. The "Deep Learning Hypothesis" is that any simple perceptual task that a human can solve in 0.1 seconds can be solved by a 10-layer neural network. Explain the idea behind that estimate.