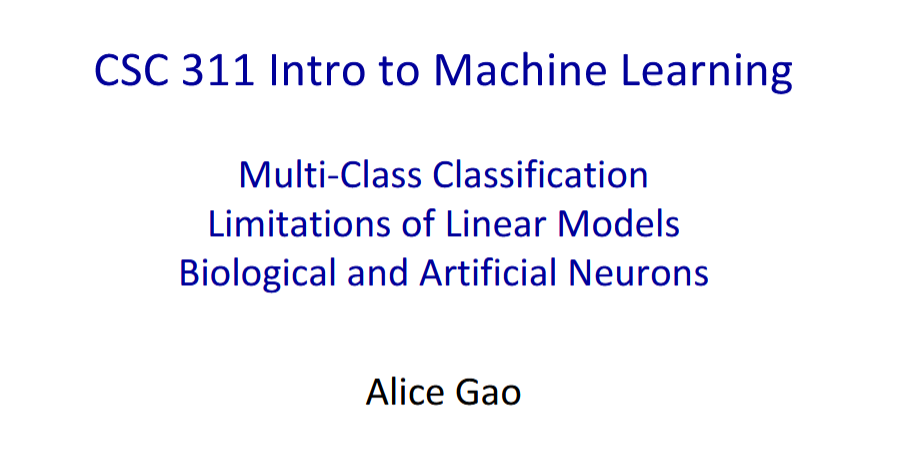
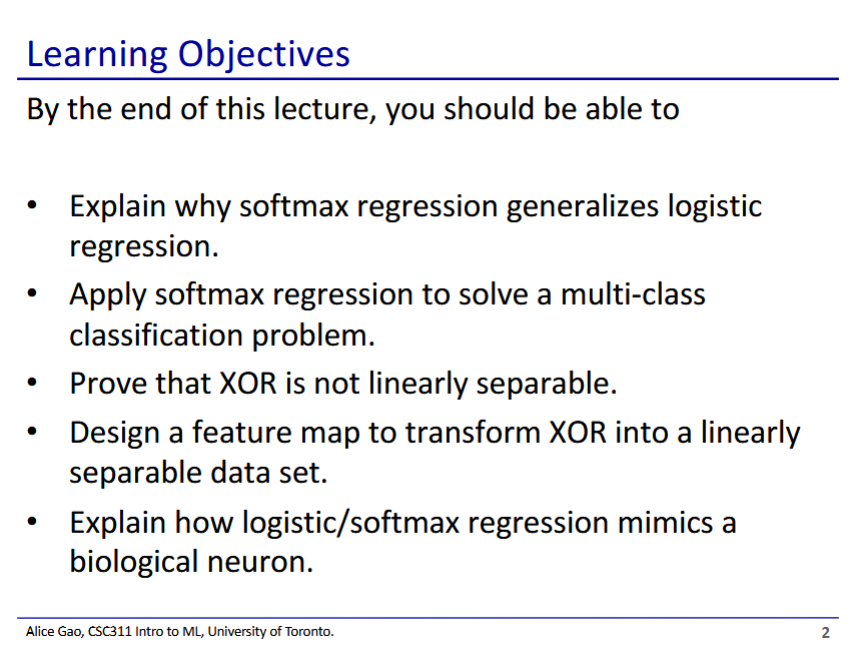
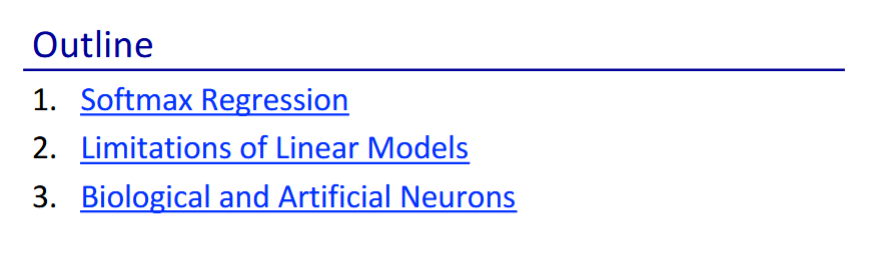
| **Admin stuff (Test things)**   * Test was too long, future tests will not be this long   + Prof. will collect midterm feedback next week along with sample solutions * Test will be graded in about a week, and marks will probably be adjusted   + Test will not affect grades in a disastrous way   **Softmax regression**   * Softmax regression is the generalised form of logistic regression   + Logistic regression classifies between 2 labels, softmax regression can classify between any k labels * Both output (**y**) and target (**t(i)**) are vectors with k components (where k is the number of possible labels)   + **t** is a one-hot vector, where only one component (the predicted label) is set to 1, and all others are set to 0   + y is the probability distribution over the k classes, the components of y sum to 1   + Recall logistic regression where y and are scalars (2 possible labels) * The linear model () now also produces a vector of k components   + W is now a matrix with dimensions (D+1) \* k * y is calculated from z using the **softmax activation function**    + , where   + If one is much larger than the rest, then softmax approaches argmax   + The softmax activation function is the generalised form of the logistic activation function, you can convince yourself of this by simplifying it for k=1   + The softmax function is similar to the **argmax** function (pick the largest one and return a one-hot vector)     - But like the 0-1 activation function, argmax does not work with gradient descent * Softmax uses a generalised form of cross-entropy loss   + Non-vectorised: LCE(**y, t**)     - is the probability of the kth label in the one-hot vector     - is the target value for the kth label in the one-hot vector     - Only 1 term is nonzero, since is a one-hot vector   + Vectorised: LCE(**y, t**) = -**t**T log(**y**)   **Limitations of linear models**   * Linear models are limited in their predictive power   + Can only perfectly classify data that is linearly separable   + Logistic and softmax regression are examples of linear models * **XOR: a function that is not linearly separable**   + Proof on slide 18 * We can perfectly classify XOR (and other non-linear hypotheses) if we use feature maps   + However designing feature maps by hand for complicated data is tricky   + Instead we will use **neural networks** |
| --- |

**Quick review:**

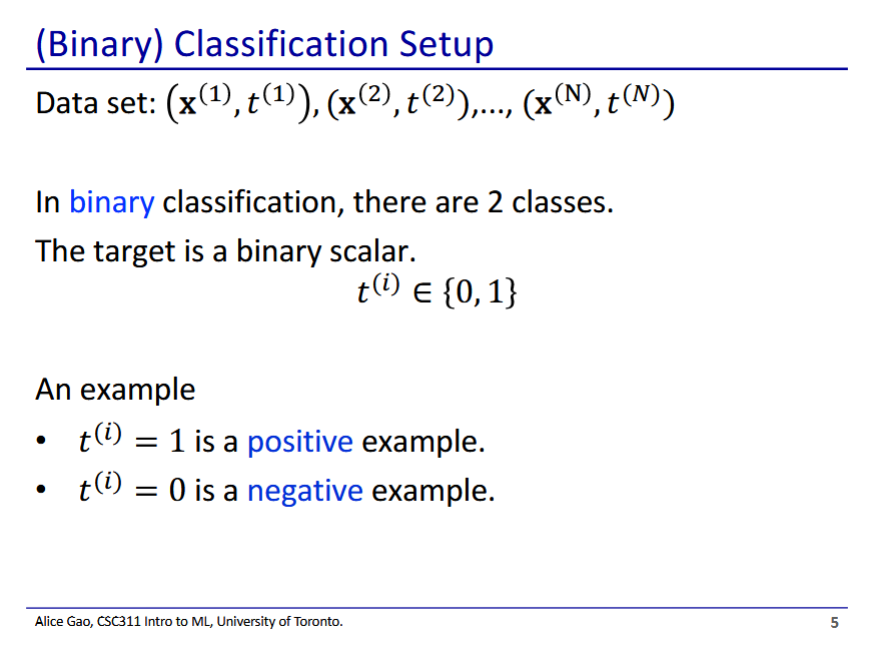
* Logistic regression
  + Linear model with sigmoid activation function
    - Sigmoid activation function gives an output between 0 and 1
  + Loss function is cross-entropy
* Journey to logistic regression
  + Why are we using the sigmoid activation function and cross-entropy loss?
* Batch gradient descent
  + Taking the full gradient descent over all data points is time-consuming if we have a lot of training data points
  + Stochastic gradient descent
    - We use 1 random data point to calculate gradient each step
    - Each update is very fast, but highly variable
    - Over a very long time we will eventually go in the right direction
  + Mini-batch gradient descent
    - We use a subset of data points to calculate gradient each step
    - We shuffle and split data into batches, and iterate through the mini-batches each update
    - Larger batch will make update slower, but more accurate to the true gradient and less variable
  + One disadvantage of batch is that we cannot vectorise our operations



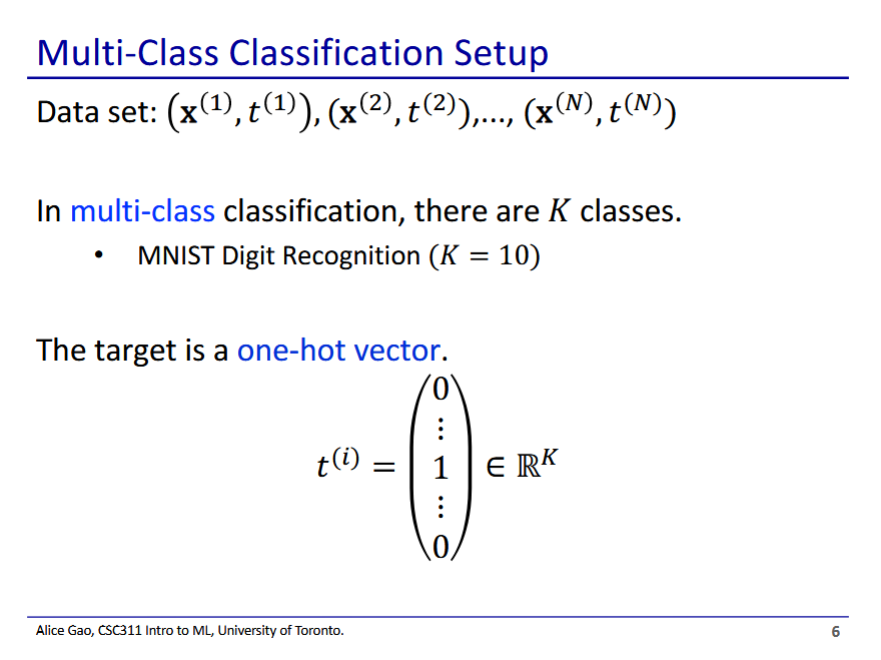




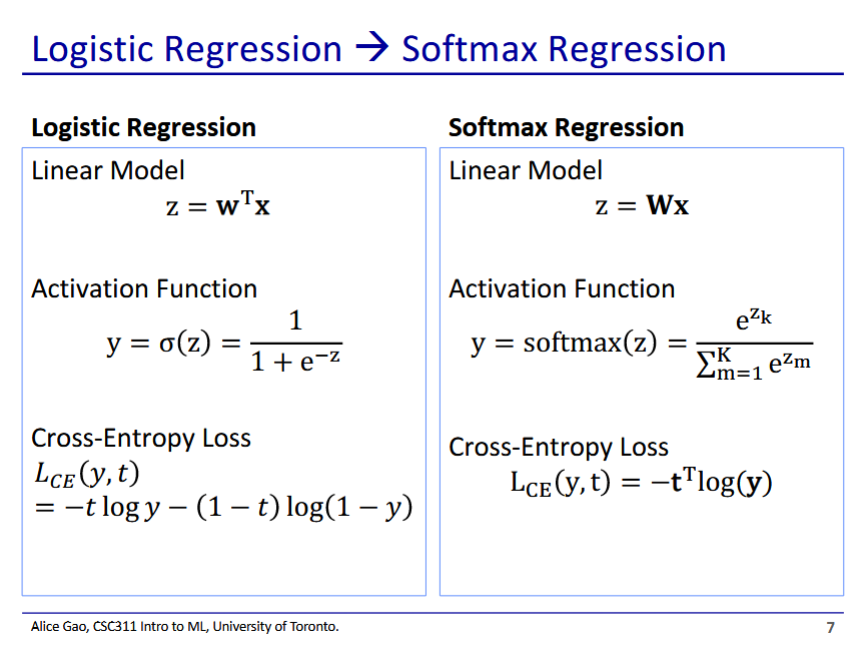




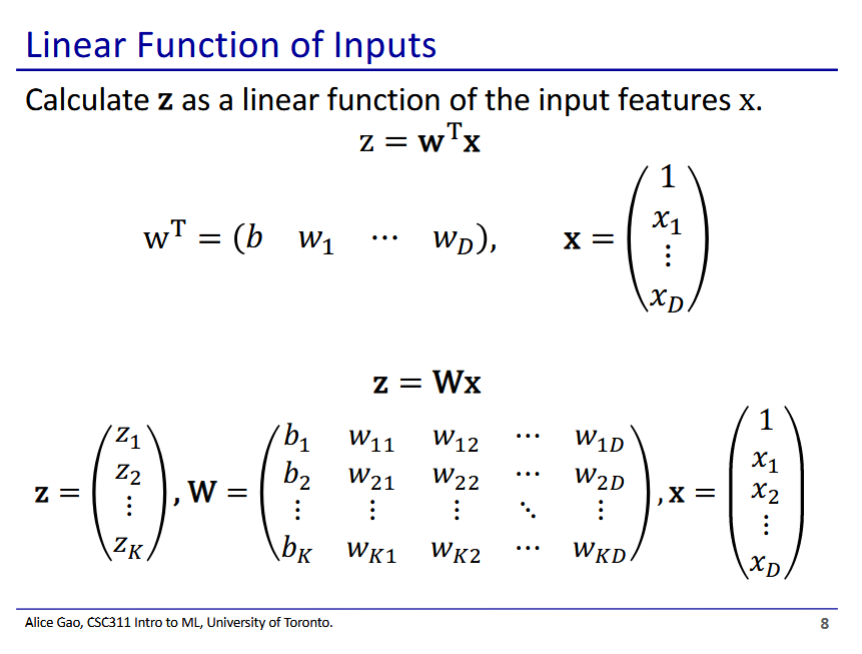
* Logistic regression is actually a classification model
  + Linear regression is a regression model (continuous numerical output), but logistic regression is a classification model (goes from 1 to 0)
  + Logistic regression is named confusingly due to historical reasons



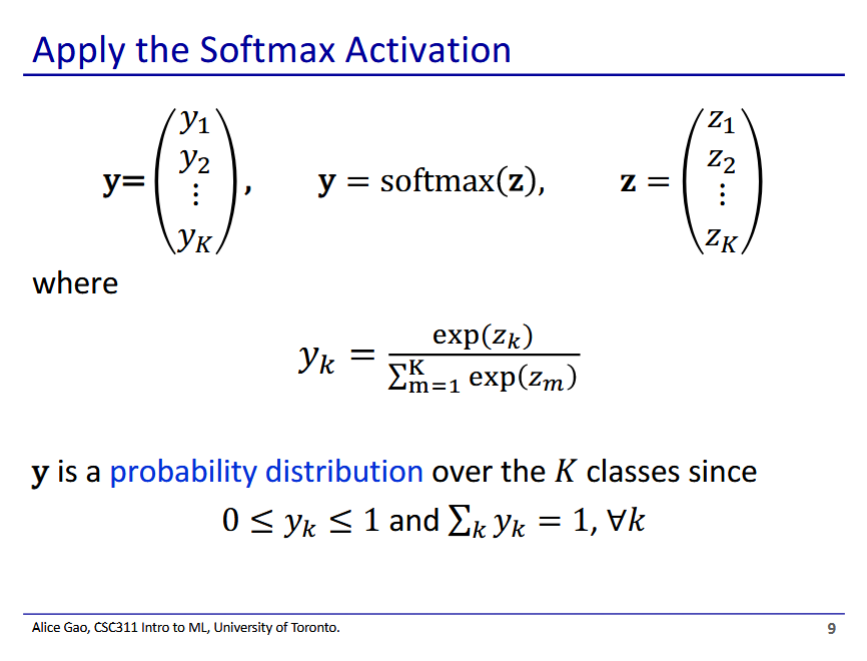
* One-hot encoding - each possible label for the output is represented as 1 component of the output vector
  + The component for predicted label is set to 1, all other components are set to 0



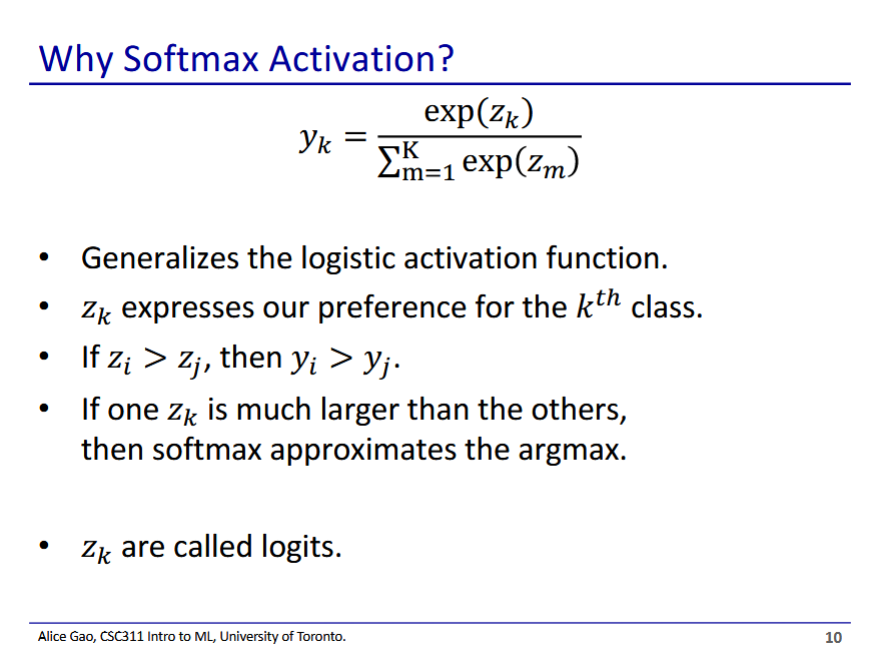
* Softmax regression is a generalisation of the logistic regression model
* Differences
  + Softmax has weights in a matrix
  + Softmax uses the softmax activation function



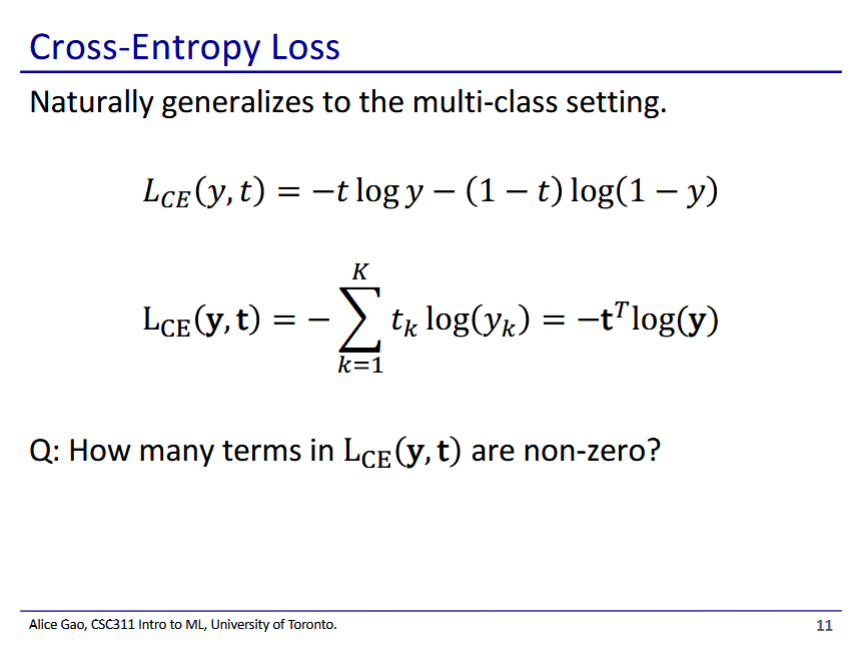
* In logistic regression, z is a scalar
* In softmax, z is a vector
* Weight matrix of softmax
  + Each row looks like the weights in logistic regression
  + We have k rows that correspond to the k components of the z vector



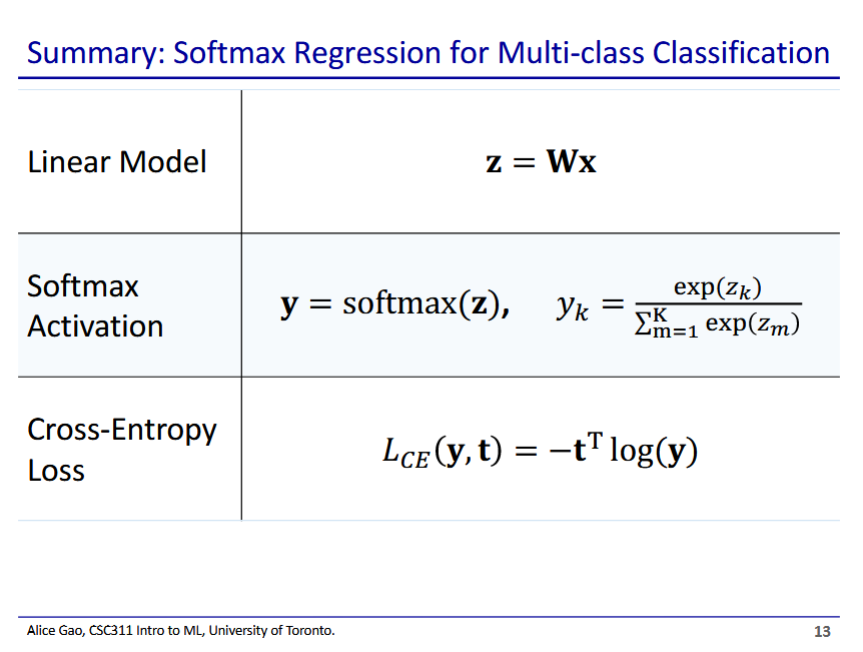
* We then apply the softmax function to z to get the final output
  + We use each z to calculate the power of the exponential
  + Then we normalise to get a probability distribution that the output belongs to the class



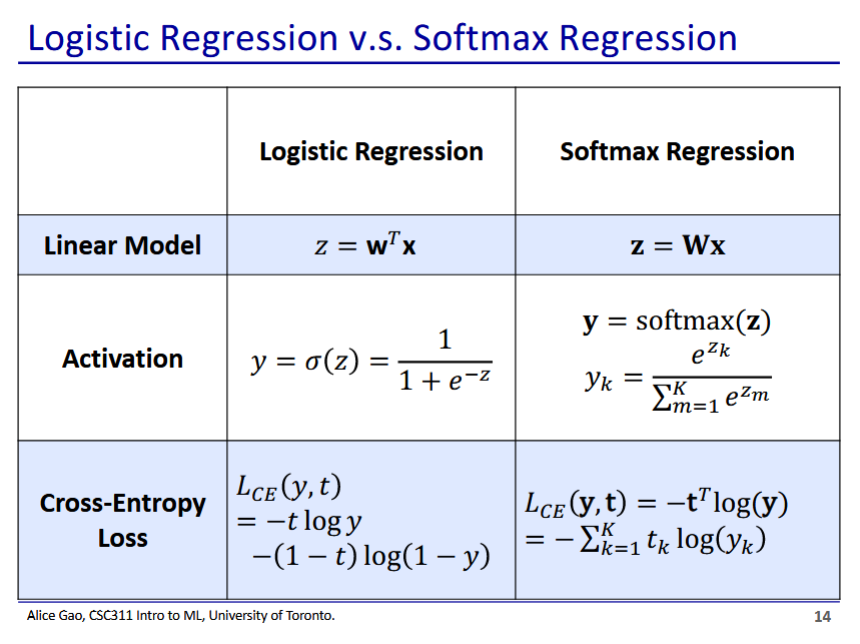
* **Exercise**: prove that softmax is the generalised logistic activation function
  + Take softmax, restrict it to 2 cases, and show that it is equivalent to logistic activation function
* Softmax is the softer version of argmax
  + Argmax picks the largest of the z components
    - Argmax is stepwise, has no gradient to descend
  + Softmax considers all the z components in final output

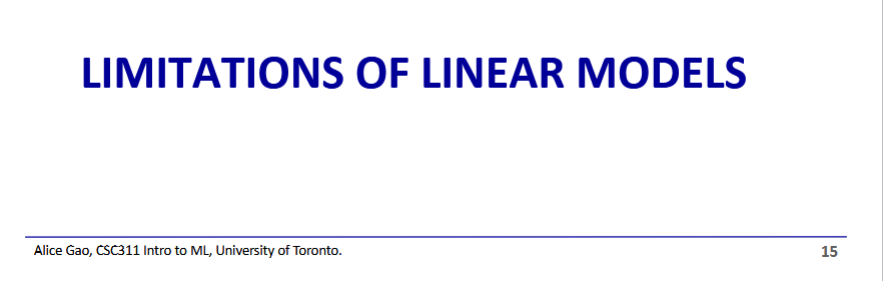


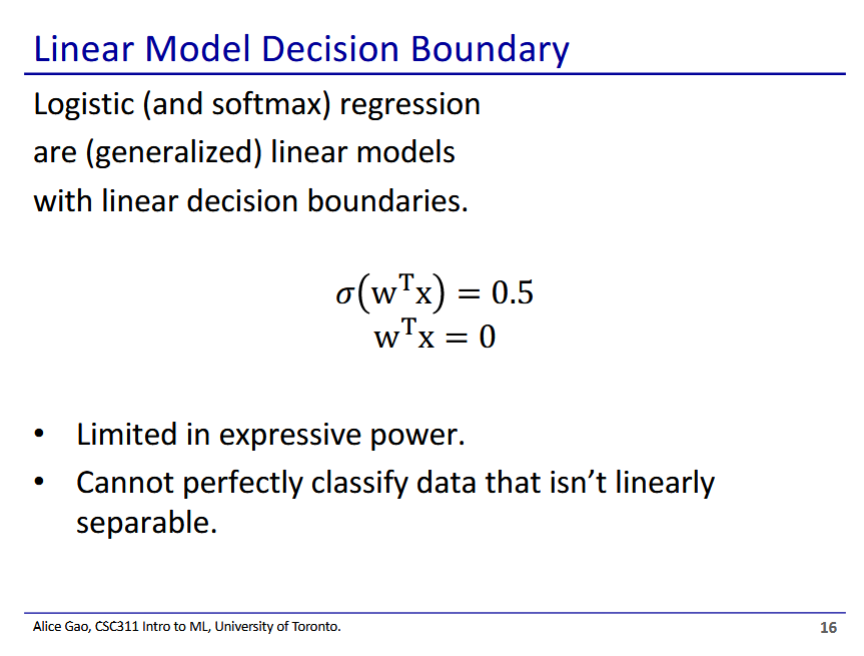
* Cross-entropy also has a more general form
* Interpreting the general form
  + is the probability of the kth label
  + is the target value for the kth label
* Vectorised form
  + Dot product between the target vector and the log of the output (probability distribution) vector
* How many terms are non-zero?
  + 1 term, since the target vector t is a one-hot vector (all values are 0 except 1)



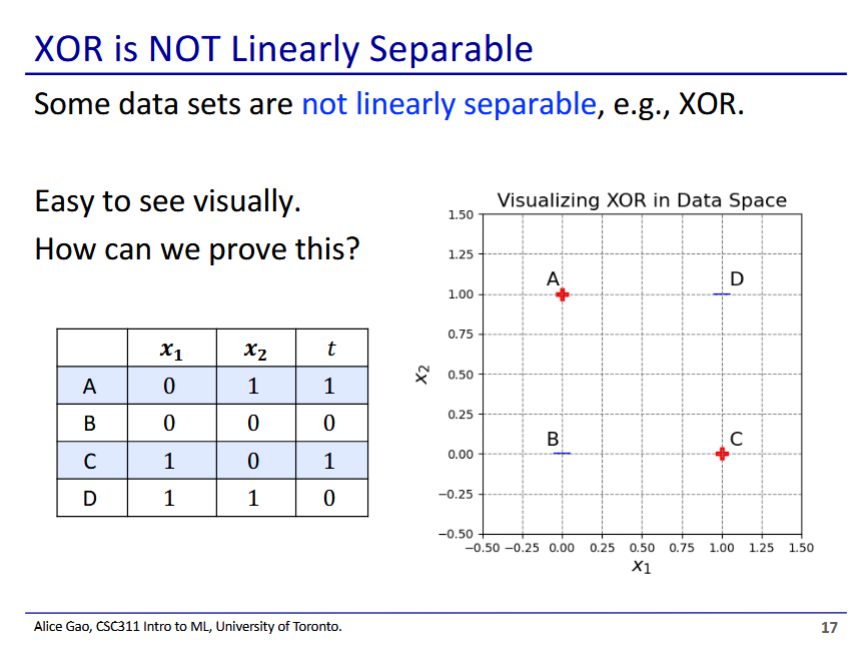
* Softmax lets us classify for k different labels
  + When k=2, softmax becomes logistic regression
* Final output is y, which is a scalar
  + However, it is important that we use the softmax activation since it has a gradient



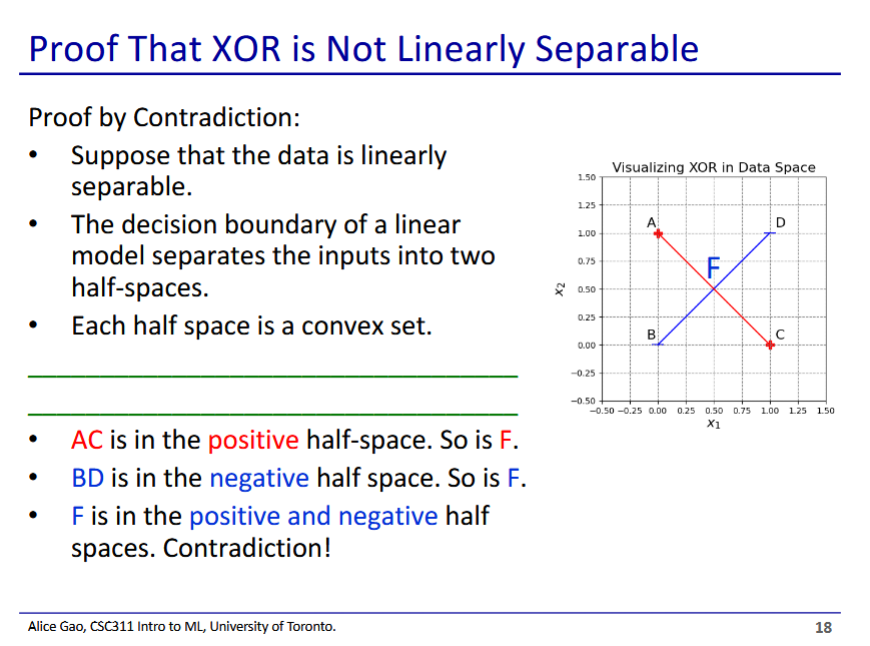




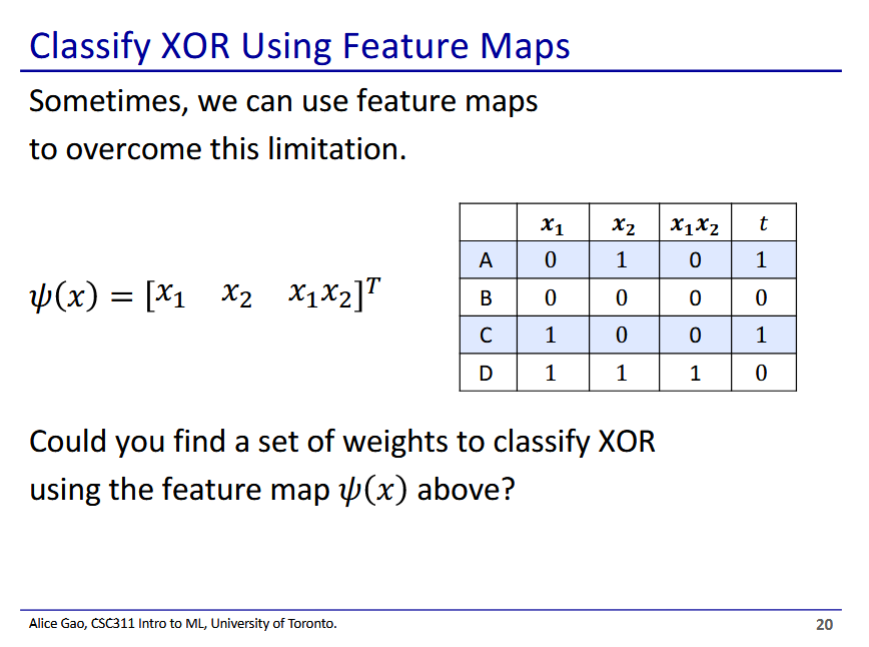
* Linear models have a significant limitation - their data must be linearly separable



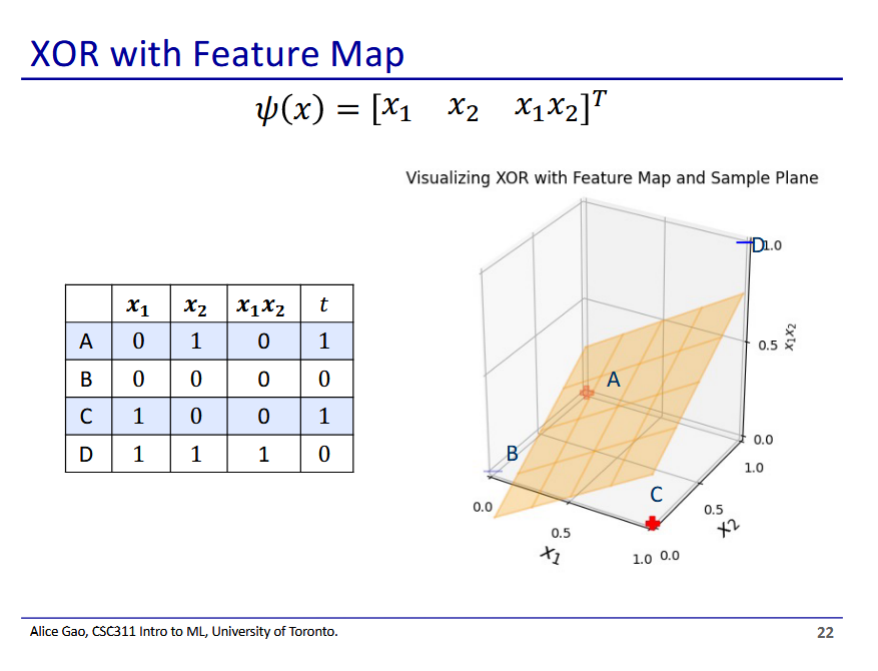
* XOR was a huge problem for machine learning historically
  + The XOR function could not be modelled by perceptrons
  + This led to the first AI winter
* XOR function:
  + Return 1 if the two discrete features are different, return 0 if the features are the same
  + Not linearly separable - you cannot draw a line that separates the + from the -



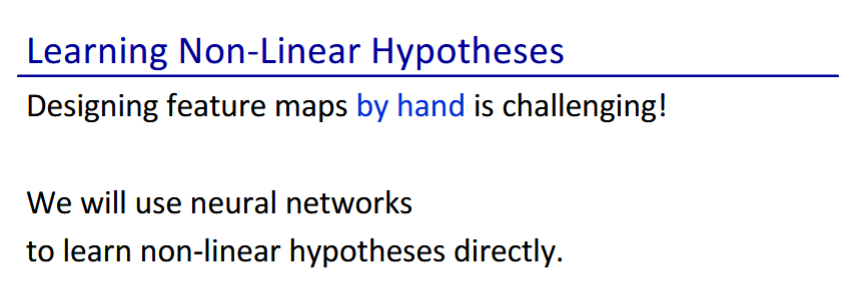
* Blank: By definition of convex set, all line segments between points on a convex set are also on the convex set



* If we add an additional feature we can classify XOR
* The things asking on the slide is probably too hard, but will show anyways



* The last feature () raises point D, allowing us to draw a plane that separates the points



* We can classify non-linear data using feature mapping
  + But designing maps is really hard
* We will need 2 layers of neural networks to learn it
  + Perceptron is 1 layer, softmax regression is 1 layer
  + Next class we will see 2 layers

