CS540 Introduction to Artificial Intelligence
Lecture 4

Young Wu
Based on lecture slides by Jerry Zhu, Yingyu Liang, and Charles Dyer

June 24, 2021

Quiz and Discussion

Admin

- Results of the two-thirds of the average game are posted on Q2 and Q3 pages, the average decreased from 20 to 15.
- No quiz questions today.
- Discussion topic on Q4 page.
- Please also volunteer to share your answers to M2 and M3 questions on Piazza, especially M2Q8 and M3Q6 Q7. Thanks!

More Practice Questions

Admin

- Math homework questions with other IDs.
- More past exam questions see W4 page.
- Last year's exams are broken, I will fix them over the weekend.

Remind Me to Start Recording

Admin

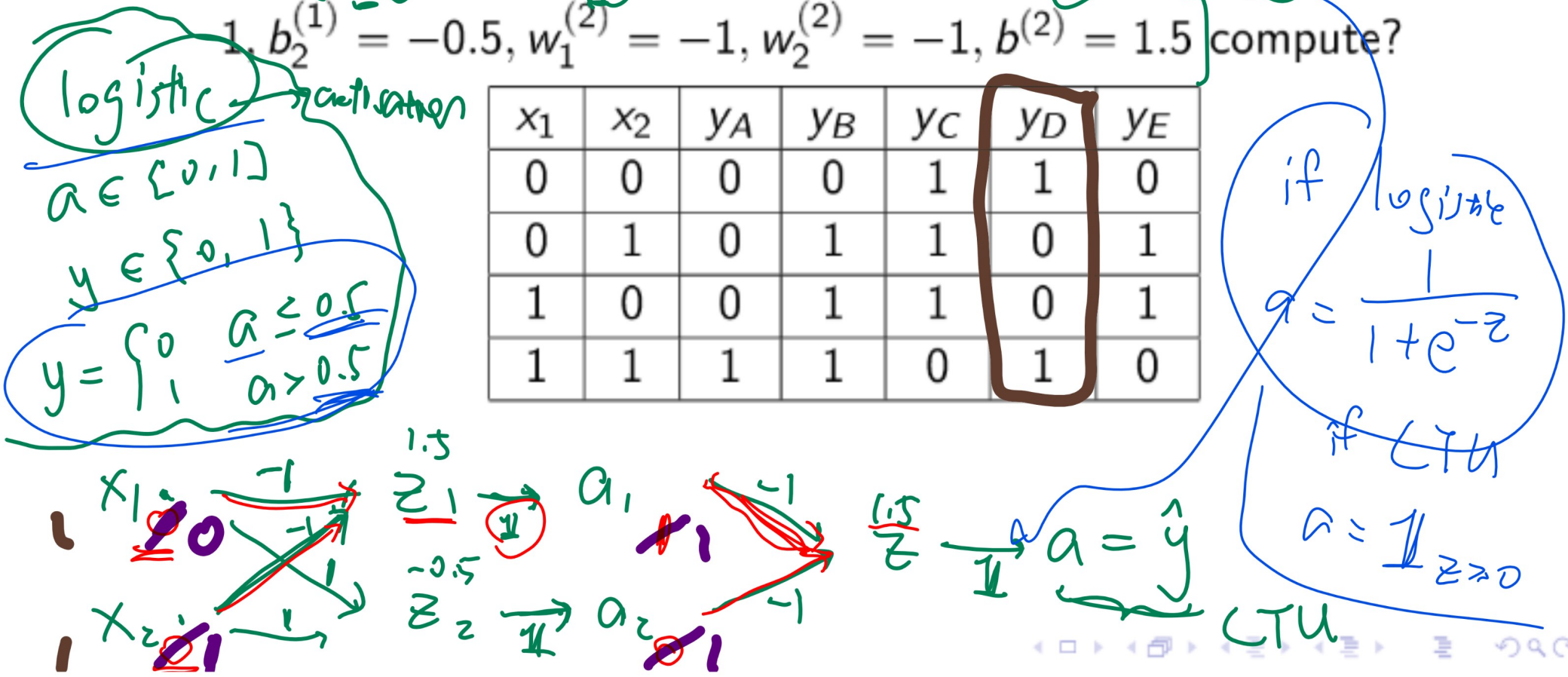
- The messages you send in chat will be recorded: you can change your Zoom name now before I start recording.

Learning Logical Operators 4

Quiz

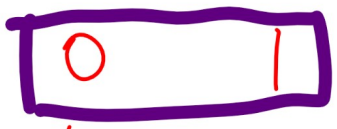
- What function does the multi-layer LTU perceptron network with $w_{11}^{(1)} = -1, w_{21}^{(1)} = -1, b_1^{(1)} = 1.5, w_{12}^{(1)} = 1, w_{22}^{(1)} = 1, b_2^{(1)} = -0.5, w_1^{(2)} = -1, w_2^{(2)} = -1, b^{(2)} = 1.5$ compute?

x_1	x_2	y_A	y_B	y_C	y_D	y_E
0	0	0	0	1	1	0
0	1	0	1	1	0	1
1	0	0	1	1	0	1
1	1	1	1	0	1	0



Learning Logical Operators 4, Answer

Quiz



x_1	x_2	z_1	a_1	z_2	a_2	z	$a = y$
0	0	1.5	1	−0.5	0	0.5	1
0	1	0.5	1	0.5	1	−0.5	0
1	0	0.5	1	0.5	1	−0.5	0
1	1	−0.5	0	1.5	1	0.5	1

Perceptron Algorithm vs Logistic Regression

Motivation

- For LTU Perceptrons, w is updated for each instance x_i sequentially.

$$w = w - \alpha (a_i - y_i) x_i$$

geometric

not $\sum_{i=1}^{n}$

- For Logistic Perceptrons, w is updated using the gradient that involves all instances in the training data.

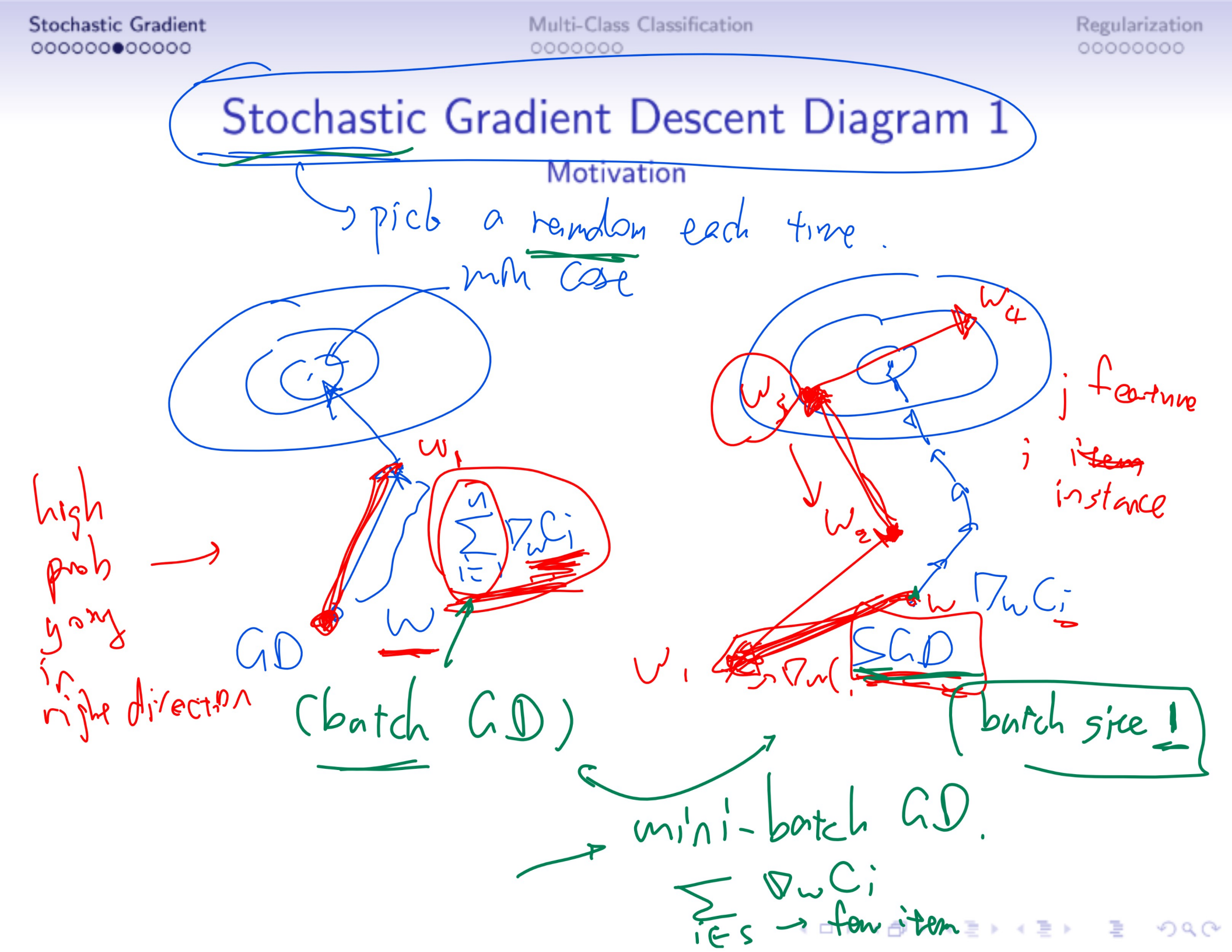
$$w = w - \alpha \sum_{i=1}^{n} (a_i - y_i) x_i$$

gradient descent

yes one item at a time.

Stochastic Gradient Descent Diagram 1

Motivation



Stochastic Gradient Descent Diagram 2

Motivation

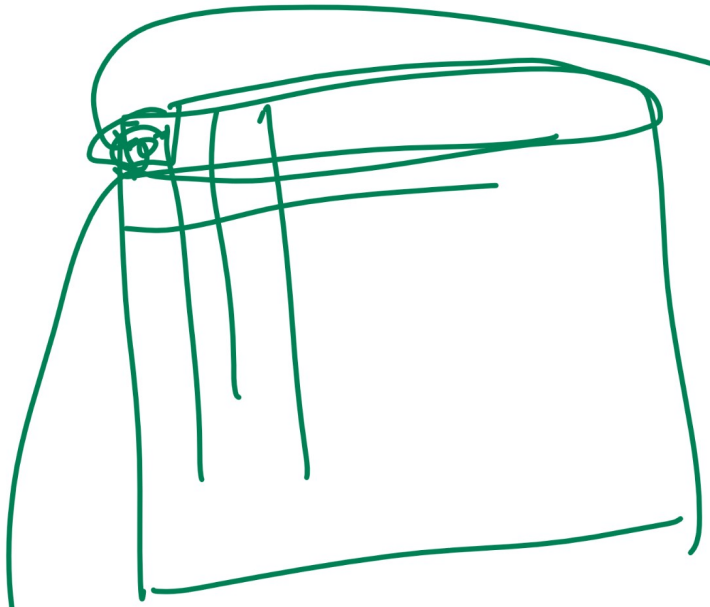
Choice of Learning Rate

Discussion

- Changing the learning rate α as the weights get closer to the optimal weights could speed up convergence.
- Popular choices of learning rate include $\frac{\alpha}{\sqrt{t}}$ and $\frac{\alpha}{t}$, where t is the current number of iterations.
- Other methods of choosing step size include using the second derivative (Hessian) information, such as Newton's method and BFGS, or using information about the gradient in previous steps, such as adaptive gradient methods like AdaGrad and Adam.

Pixel Intensity Features

Admin



R, G, B → $\frac{1}{3}(R, G, B)$

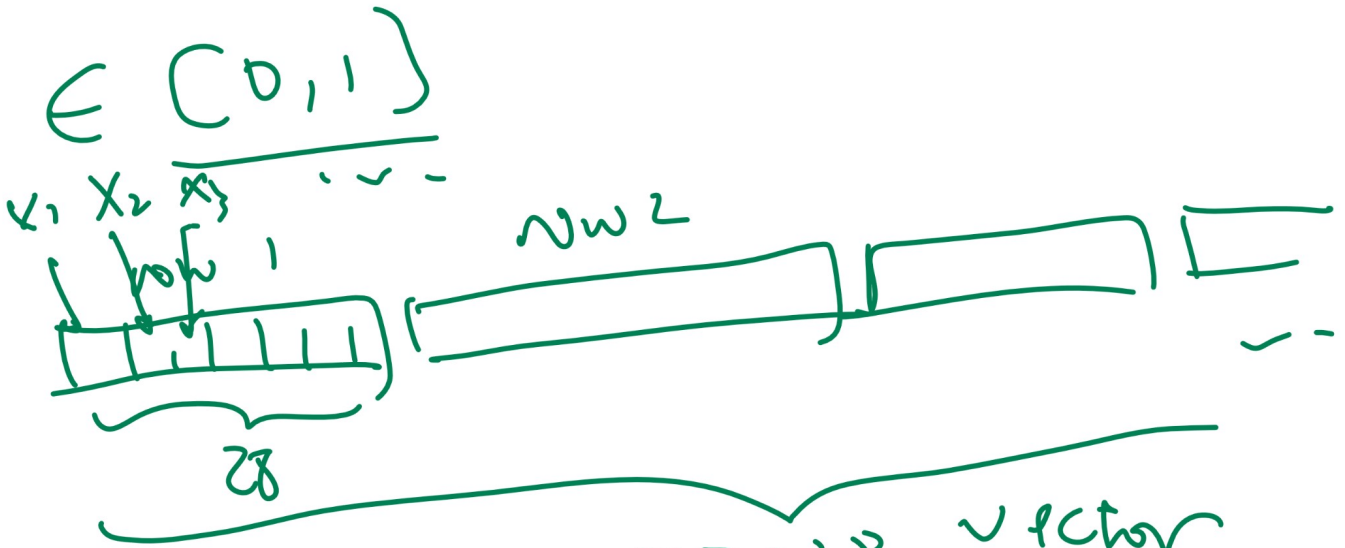
0, 255

0 − 255

255

Pixel intensity $\in [0, 1]$

flatten image



$g(w_1 x_1 + w_2 x_2 + \ldots \quad W_{28\times 28} X_{28\times 28} + b)$

Recognizing Handwritten Digits

Admin

Questions about P1

Admin

- Cost function? Any is okay.
- Learning rate? Try things.
- Stopping criterion? Discuss on Piazza (cost, gradient, max iterations).
- Stochastic vs regular gradient descent? Either.
- Regularization? If you want.
- Use test set to train? NO.
- Other questions?

Multi-Class Classification

Motivation

- When there are K categories to classify, the labels can take K different values, $y_i \in \{1, 2, ..., K\}$.
- Logistic regression and neural network cannot be directly applied to these problems.

Method 1, One VS All

Discussion

- Train a binary classification model with labels $y_i' = \mathbb{1}_{\{y_i = j\}}$ for each $j = 1, 2, ..., K$.
- Given a new test instance x_i, evaluate the activation function $a_i^{(j)}$ from model j.

$$\hat{y}_i = \arg\max_j a_i^{(j)}$$

0.7 0.6

- One problem is that the scale of $a_i^{(j)}$ may be different for different j.

Method 2, One VS One

Discussion

- Train a binary classification model for each of the $\frac{K(K-1)}{2}$ pairs of labels.
- Given a new test instance x_i, apply all $\frac{K(K-1)}{2}$ models and output the class that receives the largest number of votes.

$$\hat{y}_i = \arg\max_j \sum_{j' \neq j} \hat{y}_i^{(j \text{ vs } j')}$$

- One problem is that it is not clear what to do if multiple classes receive the same number of votes.

One Hot Encoding

Discussion

- If y is not binary, use one-hot encoding for y.
- For example, if y has three categories, then

$$y_i \in \left\{ \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right\}$$

not dummy variable

Method 3, Softmax Function

Discussion

- For both logistic regression and neural network, the last layer will have K units, a_{ij}, for $j = 1, 2, ..., K$, and the softmax function is used instead of the sigmoid function.

$$a_{ij} = g\left(w_j^T x_i + b_j\right) = \frac{\exp\left(-w_j^T x_i - b_j\right)}{\sum_{j'=1}^{K} \exp\left(-w_{j'}^T x_i - b_{j'}\right)}, j = 1, 2, ..., K$$

multi-class logistic

$$\frac{1}{1+e^{-z}}$$

$$\frac{e^{-z_1}}{e^{-z_1}+e^{-z_2}+e^{-z_3}}$$

Softmax Derivatives

Discussion

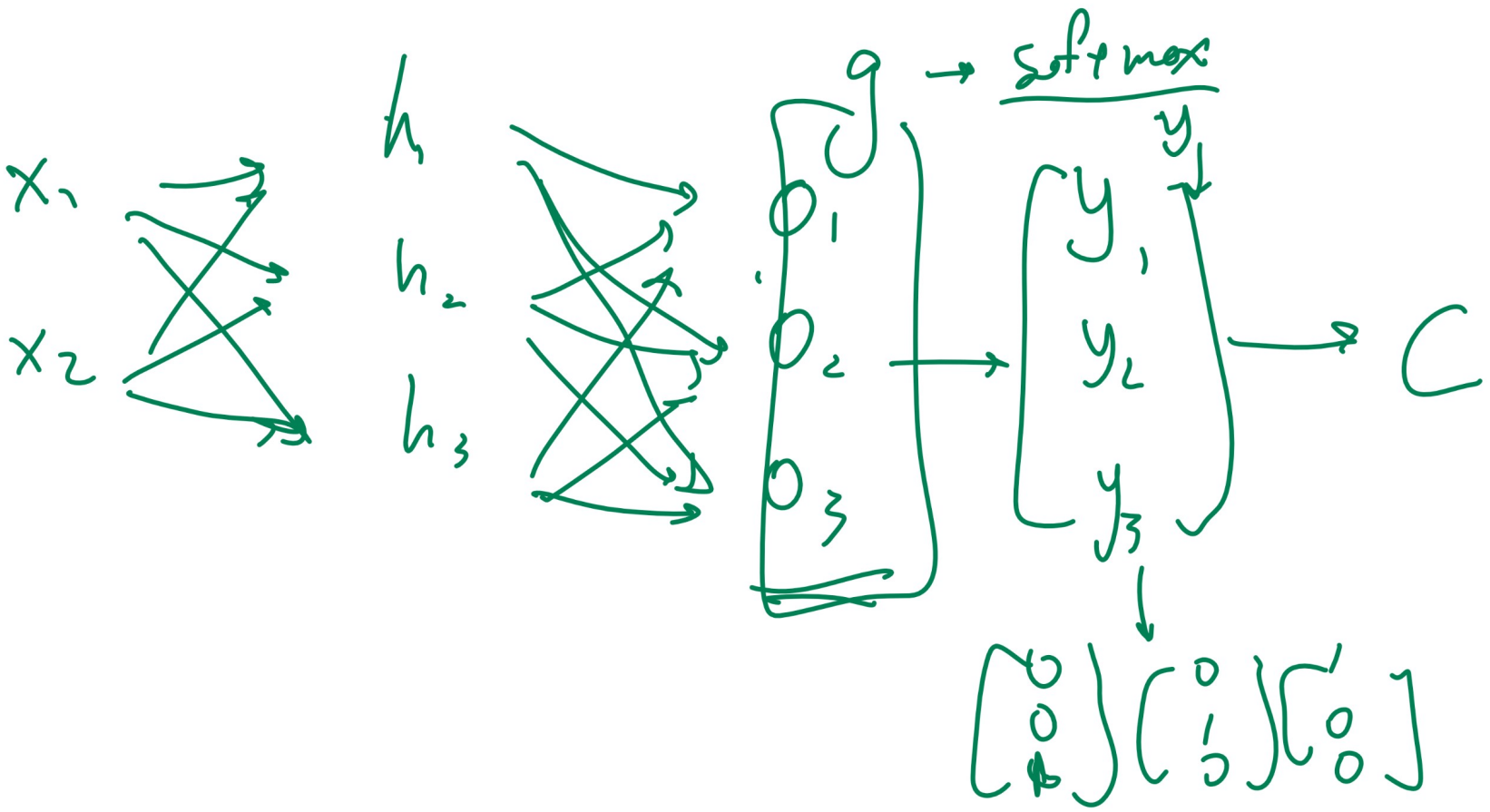
- Cross entropy loss is also commonly used with a softmax activation function.
- The gradient of cross-entropy loss with respect to a_{ij}, component j of the output layer activation for instance i has the same form as the one for logistic regression.

$$\frac{\partial C}{\partial a_{ij}} = a_{ij} - y_{ij} \Rightarrow \nabla_{a_i} C = a_i - y_i$$

- The gradient with respect to the weights can be found using the chain rule.

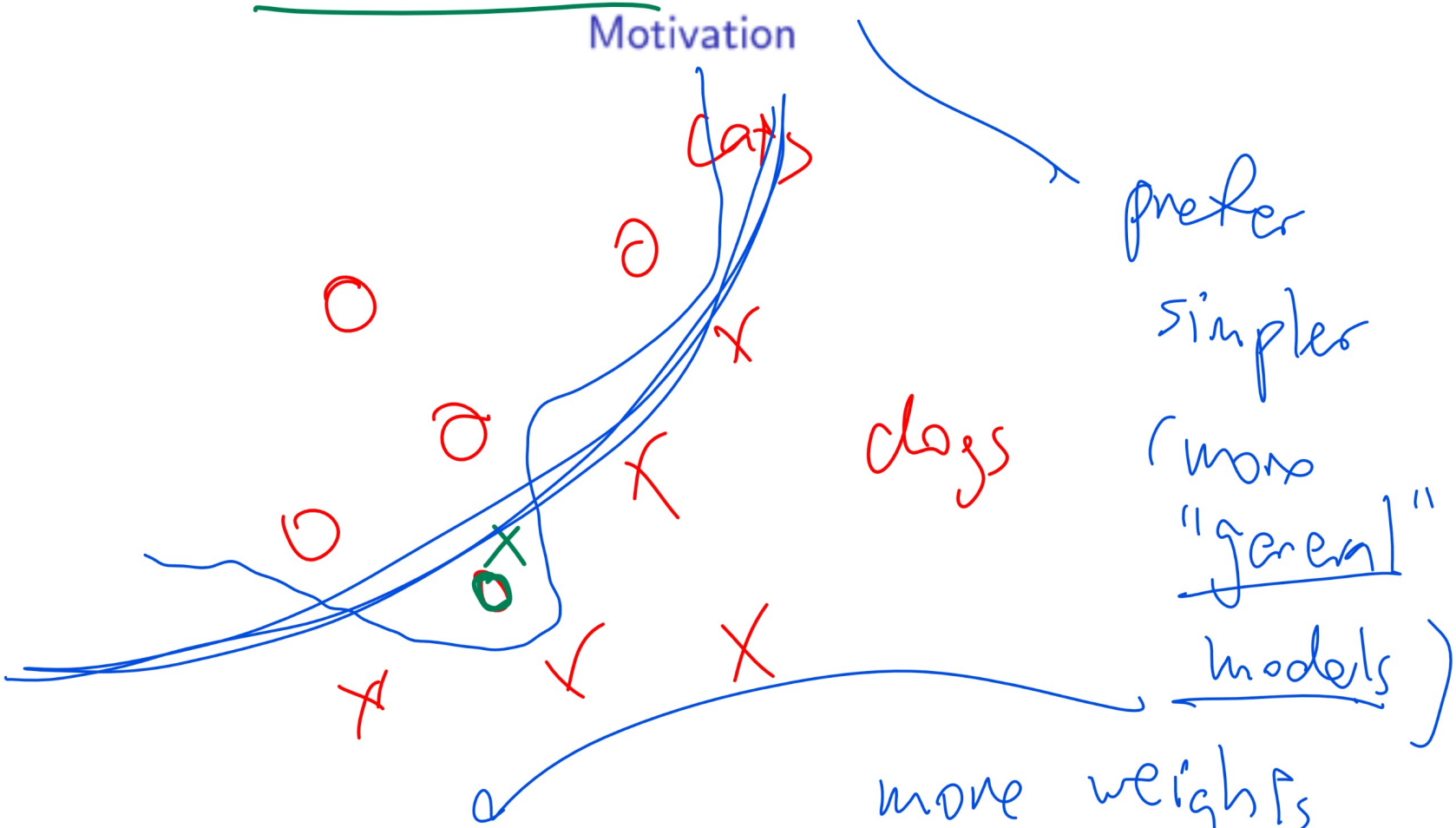
Softmax Diagram

Discussion



Generalization Error Diagram

Motivation



Method 1, Validation Set

Discussion

- Set aside a subset of the training set as the validation set.
- During training, the cost (or accuracy) on the training set will always be decreasing until it hits 0.
- Train the network until the cost (or accuracy) on the validation set begins to increase.

Method 2, Drop Out

Discussion

- At each hidden layer, a random set of units from that layer is set to 0.
- For example, each unit is retained with probability $p = 0.5$. During the test, the activations are reduced by $p = 0.5$ (or 50 percent).
- The intuition is that if a hidden unit works well with different combinations of other units, it does not rely on other units and it is likely to be individually useful.

Method 3, L1 and L2 Regularization

Discussion

- The idea is to include an additional cost for non-zero weights.
- The models are simpler if many weights are zero.
- For example, if logistic regression has only a few non-zero weights, it means only a few features are relevant, so only these features are used for prediction.

Method 3, L1 Regularization

Discussion

- For L1 regularization, add the 1-norm of the weights to the cost.

$$C = \sum_{i=1}^{n} (a_i - y_i)^2 + \lambda \left\| \begin{bmatrix} w \\ b \end{bmatrix} \right\|_1$$

$$= \sum_{i=1}^{n} (a_i - y_i)^2 + \lambda \left(\sum_{i=1}^{m} |w_i| + |b| \right)$$

- Linear regression with L1 regularization is called LASSO (least absolute shrinkage and selection operator).

many w_i exactly zero

Method 3, L2 Regularization

Discussion

- For L2 regularization, add the 2-norm of the weights to the cost.

$$C = \sum_{i=1}^{n} (a_i - y_i)^2 + \lambda \left\| \begin{bmatrix} w \\ b \end{bmatrix} \right\|_2^2$$

$$= \sum_{i=1}^{n} (a_i - y_i)^2 + \lambda \left(\sum_{i=1}^{m} w_i^2 + b^2 \right)$$

many weights close to zero

$$W = W - \alpha \nabla_W C \quad (\alpha\lambda \cdot 2W)$$

Method 4, Data Augmentation

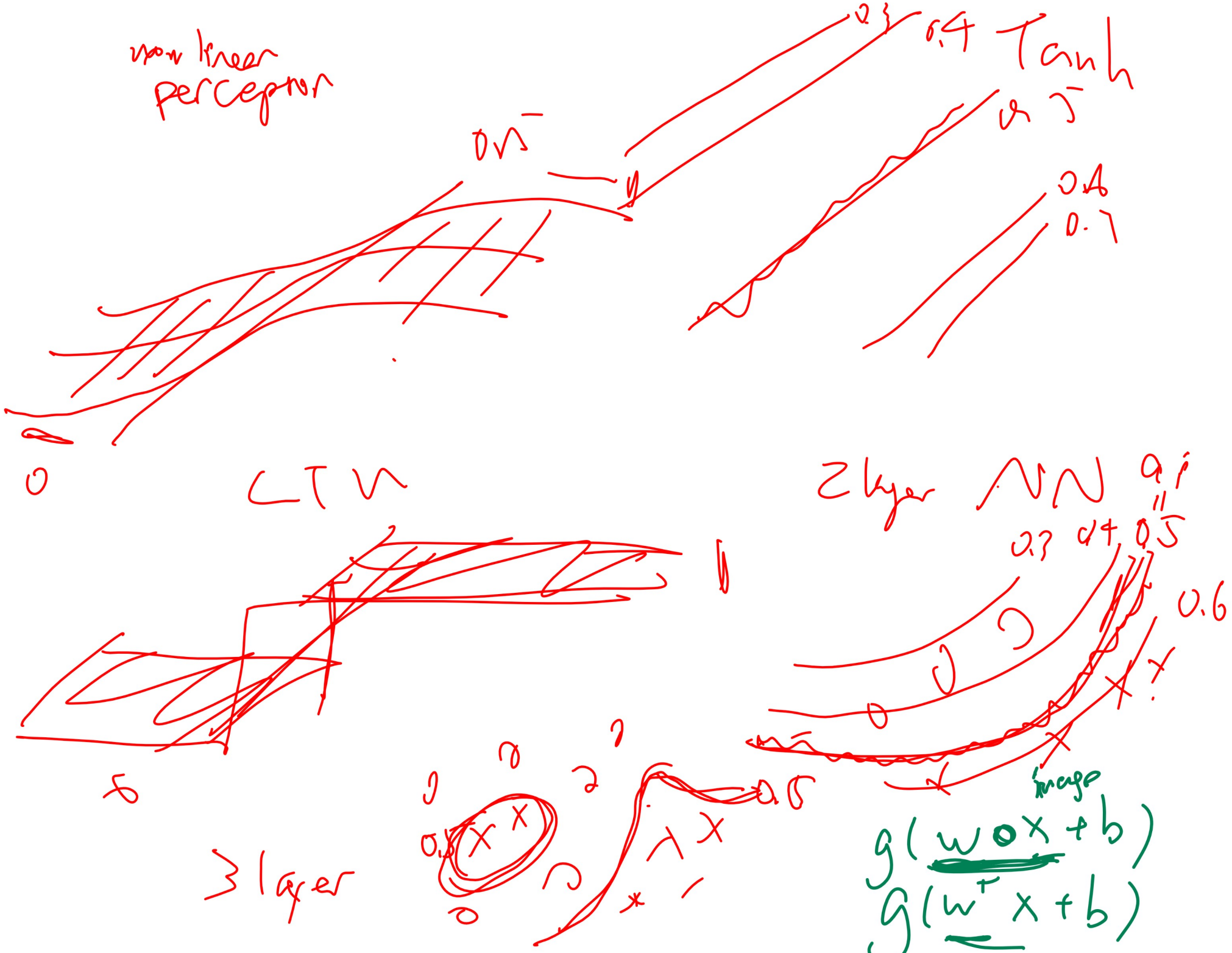
Discussion

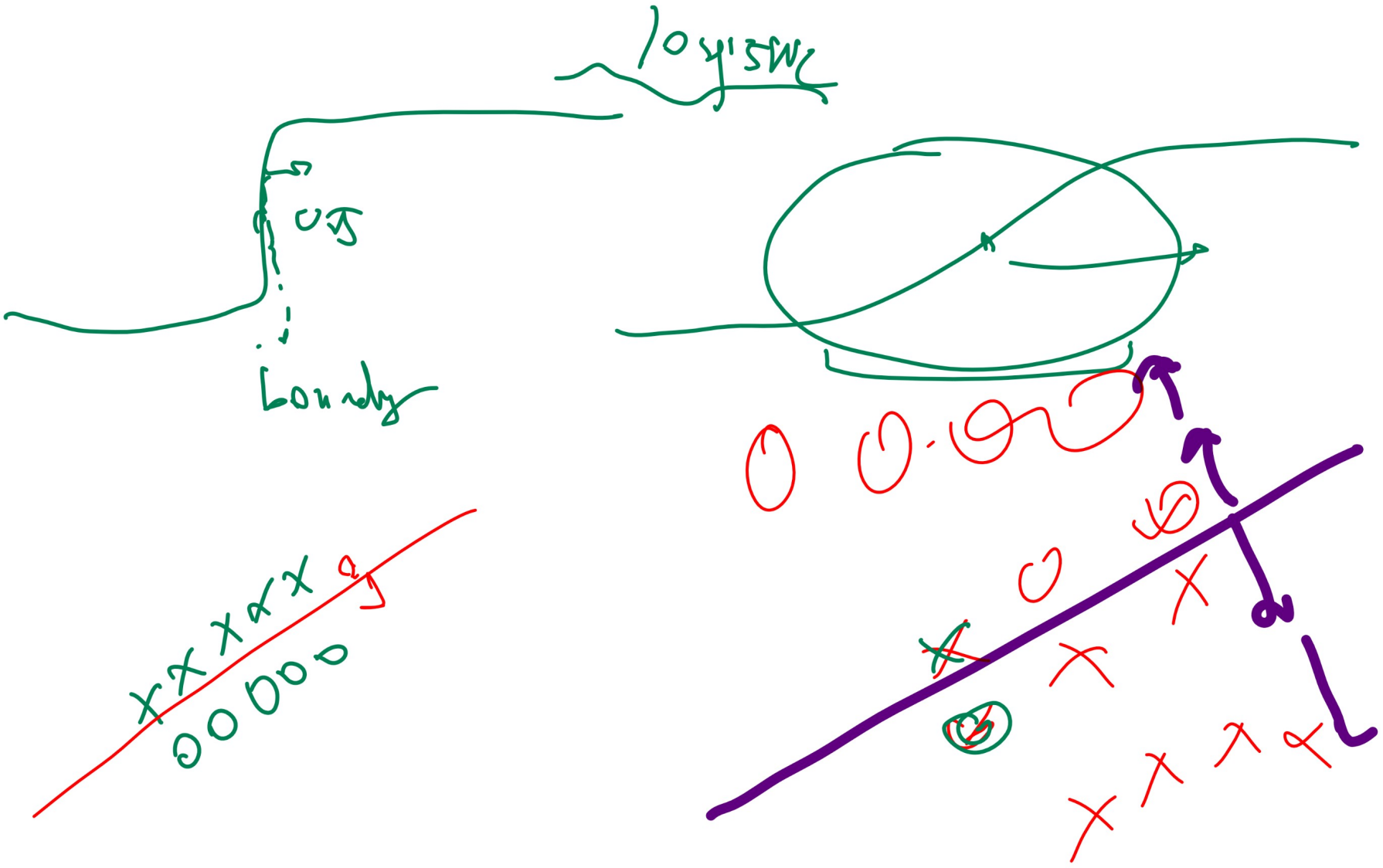
- More training data can be created from the existing ones, for example, by translating or rotating the handwritten digits.

Remind Me to Stop Recording

Admin

- If you accidentally selected an obviously incorrect answer earlier, you can enter the question name and the correct answer here.


non linear perceptron
Tanh
CTN
2layer NN
3layer
image
$g(w \odot x + b)$
$g(w^T x + b)$



Cost = 0

Cost > 0

CTN