CS540 Introduction to Artificial Intelligence

Lecture 4

Young Wu

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

July 1, 2022

Guess the Percentage

Admin

- Guess what percentage of the students (who are here) started $P1$?
- A : 0 to 20 percent.
- B : 20 to 40 percent.
- C : 40 to 60 percent.
- D : 60 to 80 percent.
- E : 80 to 100 percent.

Q1

The Percentage

Admin

Q 2

- Did you start $P1$?
- A :
- B : Yes.
- C :
- D : No.
- E :

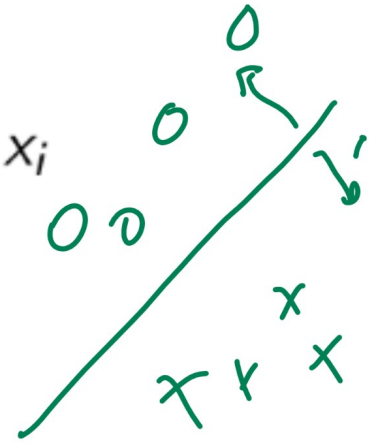
Perceptron Algorithm vs Logistic Regression

Motivation

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

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

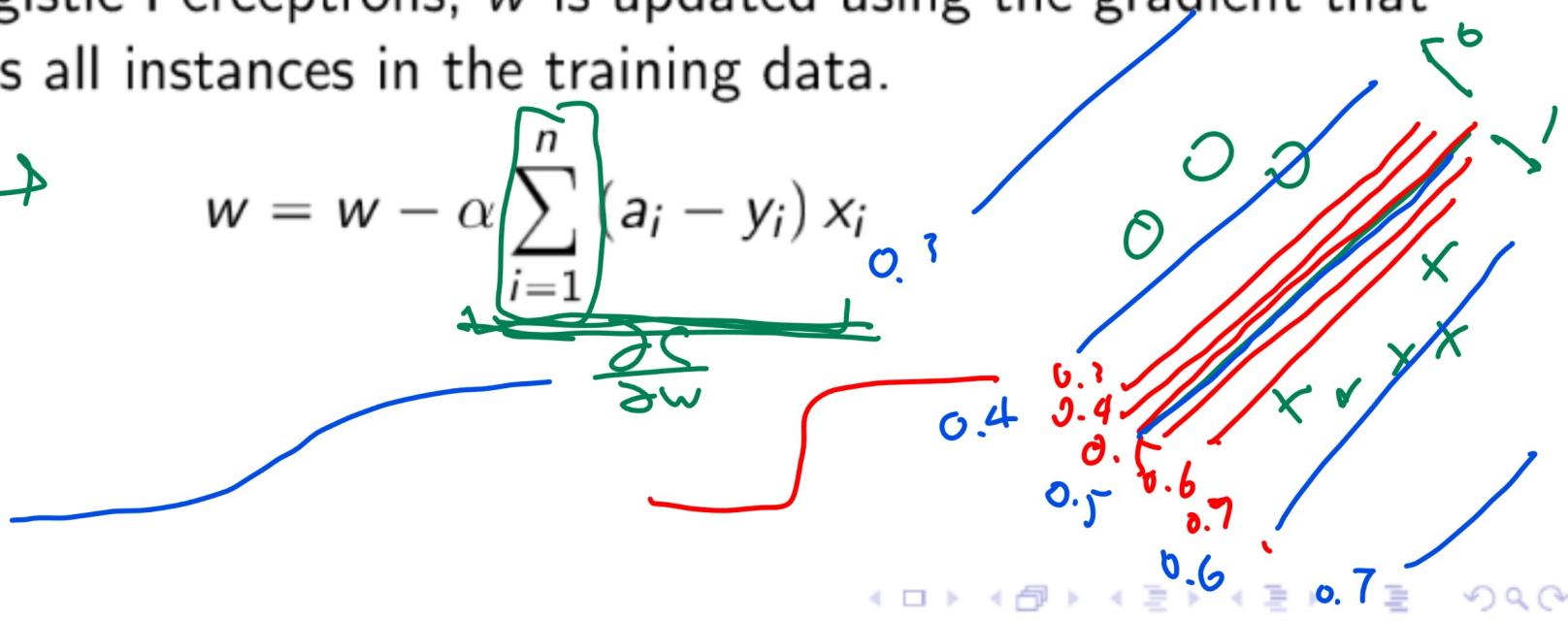
fix 1 item at a time



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

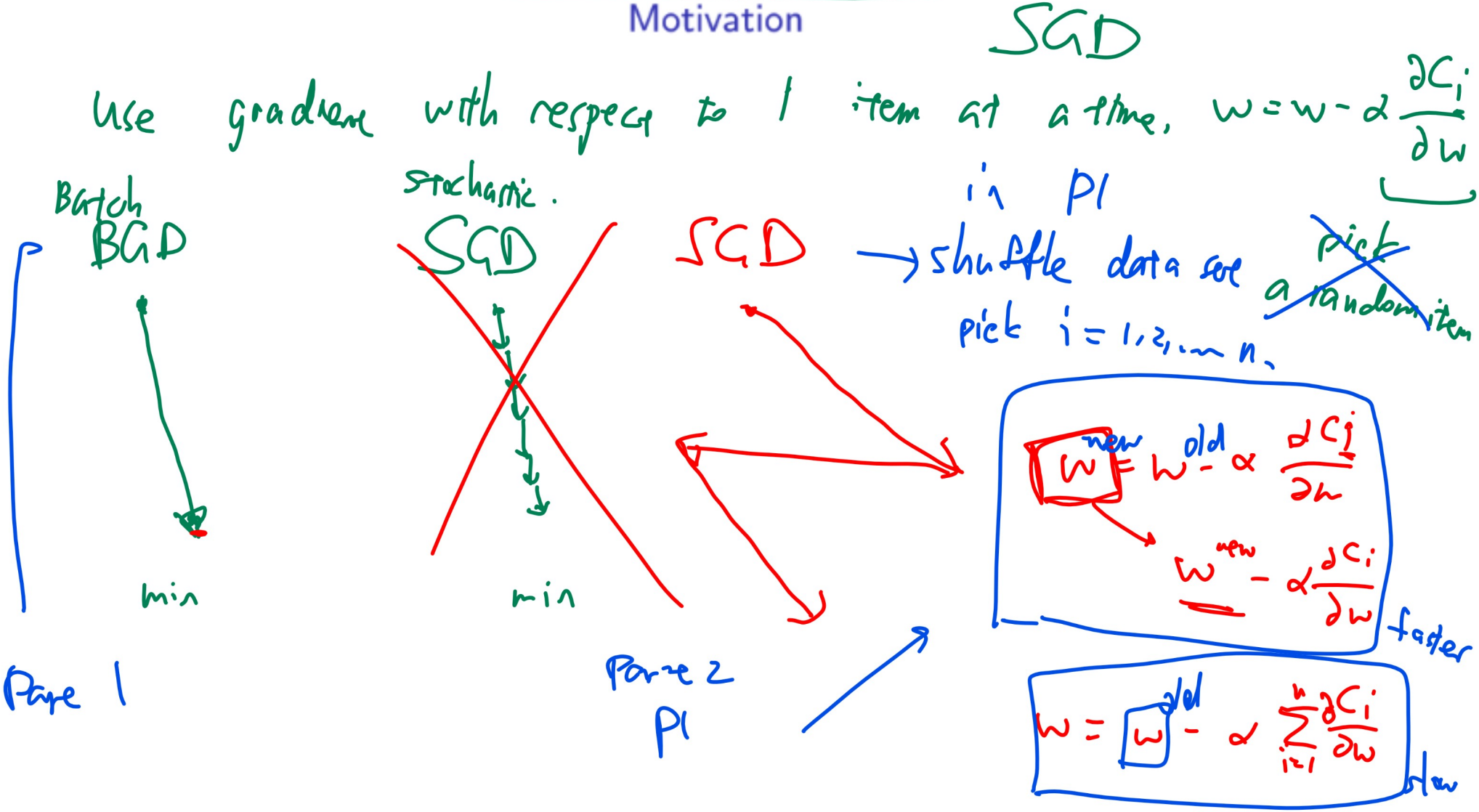
$$w = w - \alpha \sum_{i=1}^{n} (a_i - y_i) x_i$$

$\frac{\partial L}{\partial w}$



Stochastic Gradient Descent Diagram

Motivation



Stochastic Gradient

- Full gradient descent computes the gradient with respect to all instances.

$$w = w - \alpha \frac{\partial C}{\partial w} = w - \alpha \sum_{i=1}^{n} \frac{\partial C_i}{\partial w}$$

full gradient

- Stochastic gradient descent repeatedly select a random instance i and computes the gradient with respect to that instance.

$$w = w - \alpha \frac{\partial C_i}{\partial w}$$

stochastic gradient.

- Usually, random sampling is done without replacement by shuffling the training set instead of sampling with replacement, so that all instances are included in each iteration (called an epoch).

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.

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.

$$\hat{y}_i = \underset{j}{\operatorname{argmax}}\, a_i^{(j)}$$

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

Method 2, One VS One

Discussion

1 vs 2
2 vs 3
1 vs 3

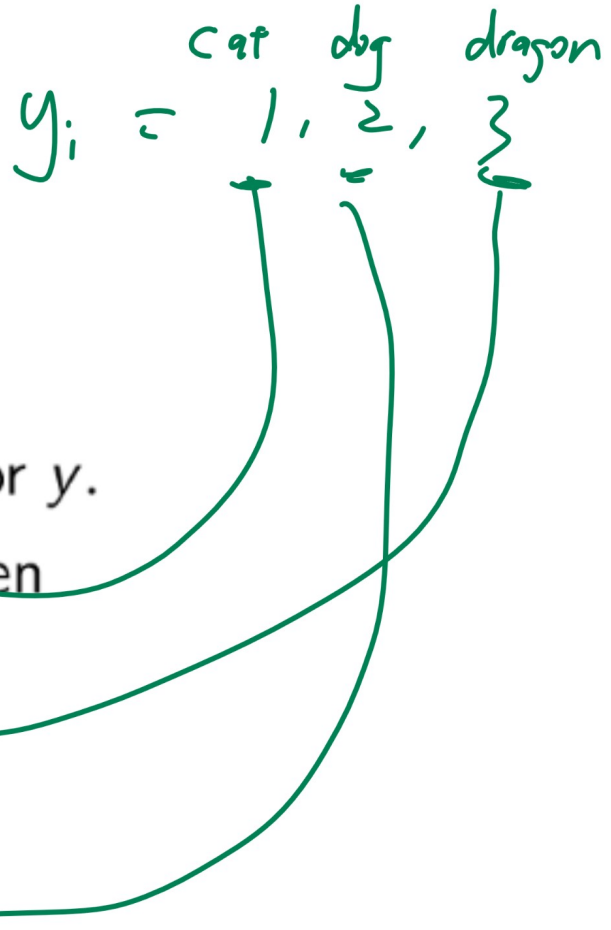
- Train a binary classification model for each of the $\frac{K(K-1)}{2}$ pairs of labels.
- Given a new test instance x_i, apply all $\frac{K(K-1)}{2}$ models and output the class that receives the largest number of votes.

$$\hat{y}_i = \underset{j}{\operatorname{argmax}} \sum_{j' \neq j} \hat{y}_i^{(j \text{ vs } j')}$$

- 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

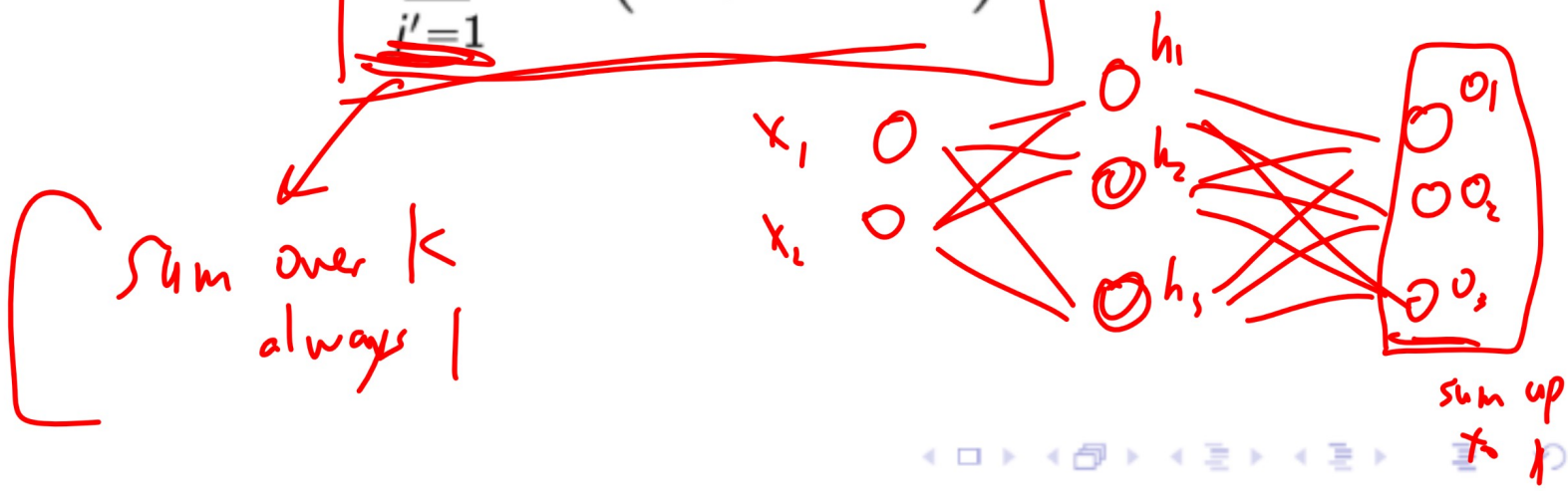
$$y_i \in \left\{ \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right\}$$

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.

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



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.

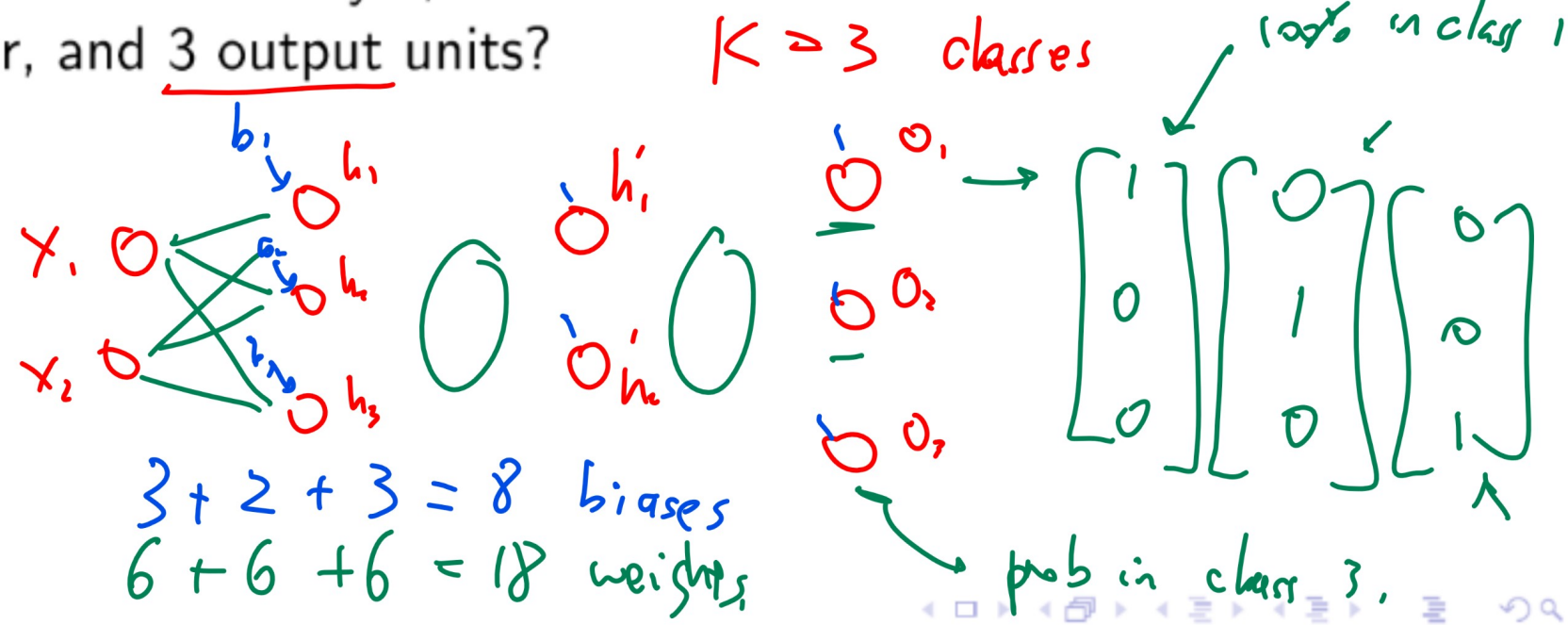
Softmax Diagram

Discussion

Weight Count

Quiz

- How many weights and biases are there in a (fully connected) three-layer neural network with 2 input units, 3 hidden units in the first hidden layer, 2 hidden units in the second hidden layer, and 3 output units?

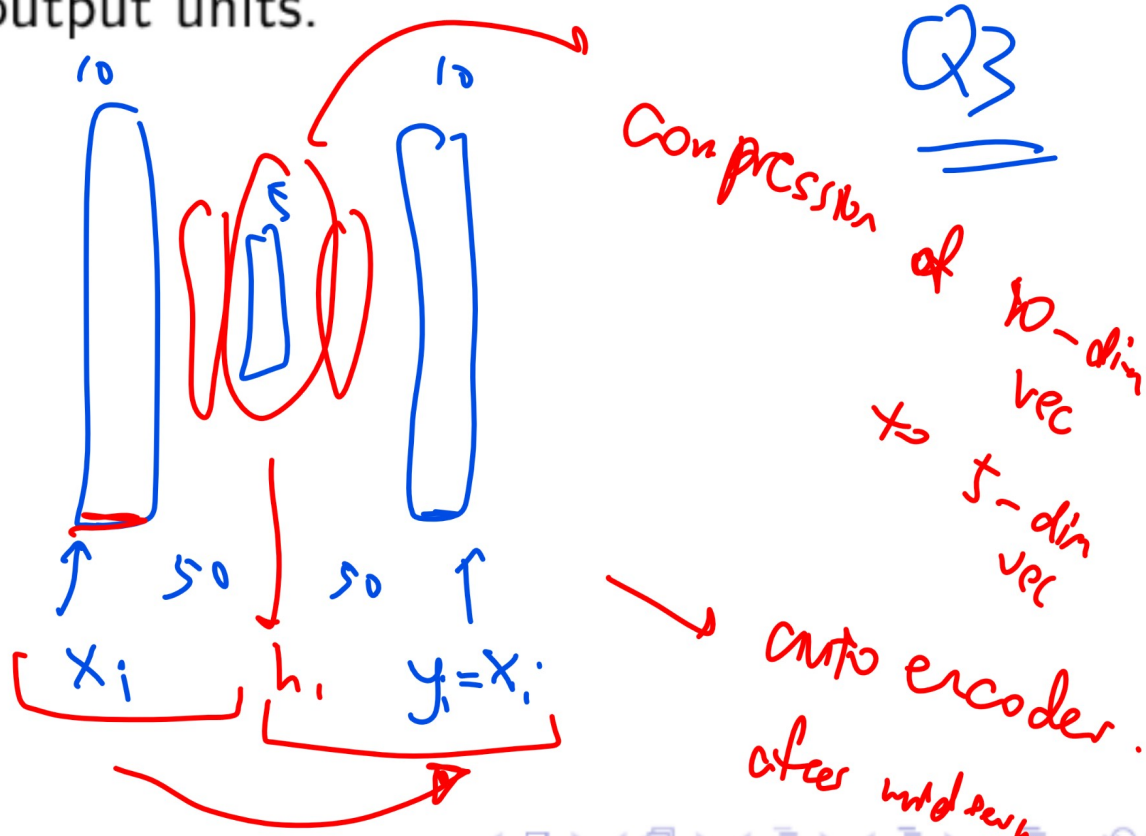


Weight Count 2

Quiz

- How many weights (not including bias) are there in a (fully connected) two-layer neural network with 10 input units, 5 hidden units, and 10 output units.

- A : 50
- B : 55
- C : 100
- D : 110
- E : 500



Weight Count 3

Quiz

- How many biases are there in a (fully connected) two-layer neural network with 10 input units, 5 hidden units, and 10 output units.
- A : 5
- B : 10
- C : ~~15~~
- D : 20
- E : 25

Q4.

hidden and output have bias terms.

Generalization Error Diagram

Motivation

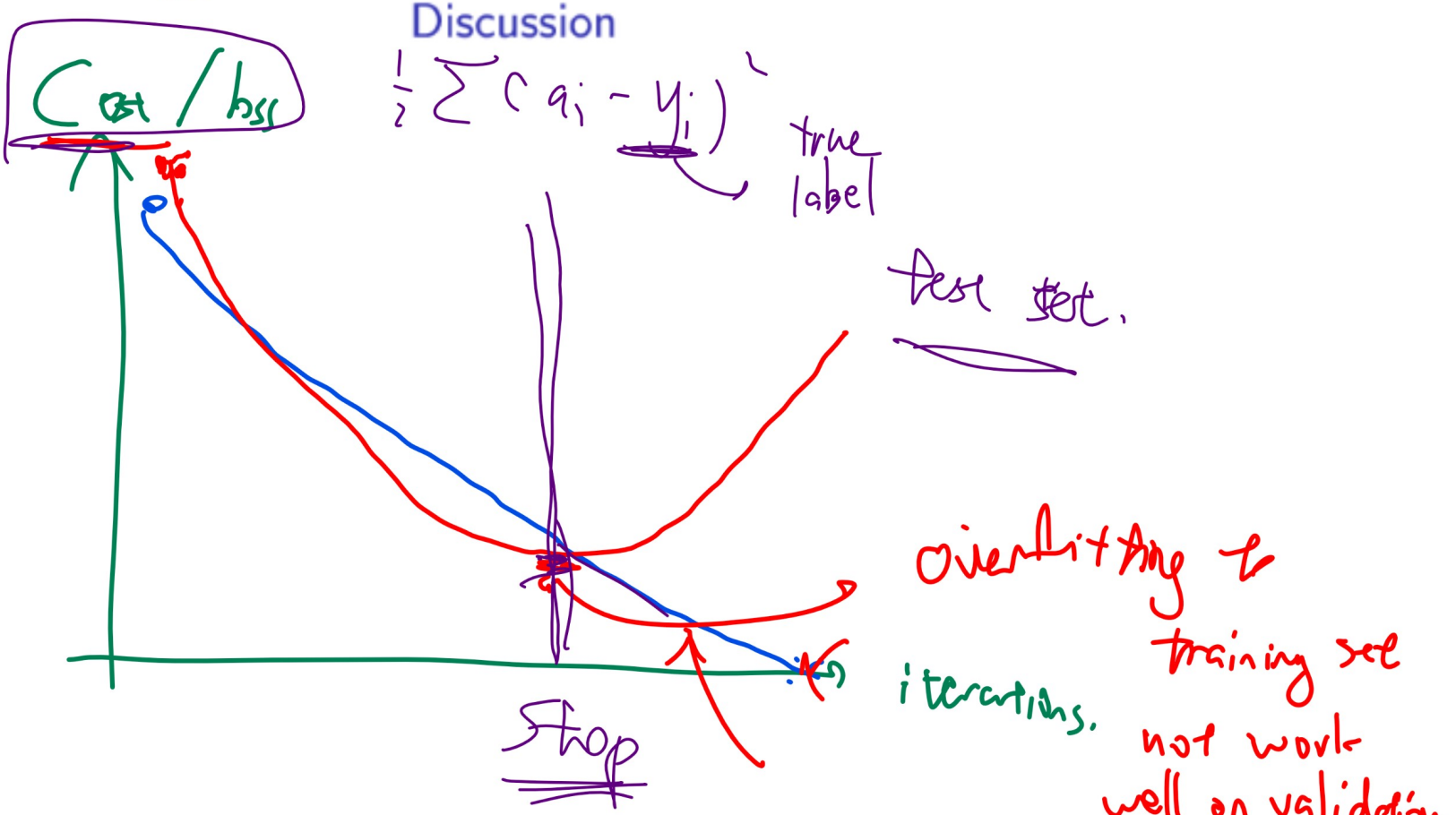
Method 1, Validation Set

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.

Validation Set Diagram

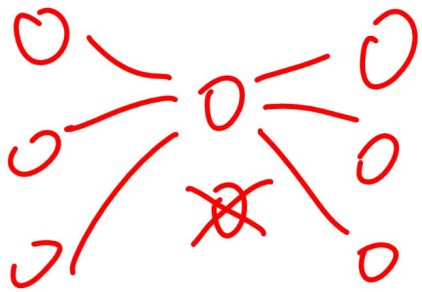
Discussion



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.



Drop Out Diagram

Discussion

Method 3, $L1$ and $L2$ Regularization

Discussion

$\min C + \lambda \|w\|$

- 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).

Method 3, $L2$ Regularization

Discussion

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

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

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

← make weights close to 0

regularization param

$$\begin{bmatrix} w := w - \frac{\partial C}{\partial w} - \lambda w \\ b := b - \frac{\partial C}{\partial b} - \lambda b \end{bmatrix}$$

small w ⇒ small cost loss

Method 4, Data Augmentation

Discussion

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

Recognizing Handwritten Digits Demo

Discussion

Questions about *P*1

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?

$$y \simeq g(w^T x + b)$$